

Relatório - Atividade T12

Introdução à Programação Paralela - M0644A

Professor: Ph.D. Guido Araujo

Aluno: Andrius H. Sperque RA: 189918

16 de Junho de 2017

1. Introdução

Este relatório tem por objetivo detalhar os resultados obtidos pela execução do experimento "T12-Paralelização com OMPCLOUD", o qual tem por objetivo paralelizar multiplicações de matrizes ($D=A.B$; $E=C.D$) na nuvem usando OmpCloud, matrizes tais que possuem tamanho variável N.

Devido ao limite de tamanho máximo para o relatório, alguns conteúdos estaram resumidos. Nos itens 2 e 3 são apresentados, respectivamente, os procedimentos realizados e os resultados obtidos.

2. Procedimentos realizados

Para alcançar os resultados esperados para este experimento, foram necessárias as seguintes atividades:

- 1) Modificação do programa serial para torná-lo paralelizável.
- 2) Configuração do Clang, específico para o OMPCloud, na área do aluno no Parsusy:
- 3) Configuração de um Cluster HDInsights com Spark na plataforma Azure da Microsoft.
- 4) Executou-se o programa serial e o programa paralelo.

Arquivo:cloud_rtl.ini.azure

```
[AzureProvider]
Cluster=acluster
StorageAccount=andriusstorage
StorageAccessKey=d2lDoqKgmiDKE8Nysi5nzSnvKLNc66qw8RgPIW60EYBVq7uA6z+KG27YoEh1O2pIMtAf8gbEuJQAk
PMuRGbXCg==
[Spark]
User=sshuser
WorkingDir=/ompccloud_test
AdditionalArgs=--executor-memory 15g --driver-memory 15g --driver-cores 8 --executor-cores 8
```

3. Resultados

3.1 Tarefa 1 - Anotar os laços do programa com as diretivas de OpenMP 4.0+. (**Programa dentro da pasta Zip**).
Compilar o programa com o clang e rodar a execução localmente.

Para N= 2000

OMPCLOUD Runtime: 19.519058s

CPU Runtime: 85.151175s

Non-Matching CPU-GPU Outputs: 0

Libomptarget --> Unloading target library!

Target Cloud RTL --> Uploading = 0s

Target Cloud RTL --> Downloading = 0s

Target Cloud RTL --> Compression = 3s

Target Cloud RTL --> Decompression = 0s

Target Cloud RTL --> Execution = 18s

Para N= 5000

MPCLOUD Runtime: 298.804782s

CPU Runtime: 2584.425932s

Non-Matching CPU-GPU Outputs: 0

Libomptarget --> Unloading target library!

Target Cloud RTL --> Uploading = 0s

Target Cloud RTL --> Downloading = 0s

Target Cloud RTL --> Compression = 18s

Target Cloud RTL --> Decompression = 0s

Target Cloud RTL --> Execution = 291s

3.2 Tarefa 2 - Criar um cluster Spark no Microsoft Azure e testar a execução na nuvem, analisar a paralelização usando a interface gráfica de profiling do Spark e modificar o tamanho das matrizes N para observar o efeito na execução .

Dashboard	Secure Shell (SSH)	Scale cluster	Move	Delete
Essentials ^				
Resource group (change) andriusresource		Cluster type, HDI version Spark 2.0 on Linux (HDI 3.5)		
Status Running		URL https://acluster.azurehdinsight.net		
Location South Central US		Learn more Documentation		
Subscription name (change) Azure Pass		Getting started Quickstart		
Subscription ID b993b808-3e78-498f-92f6-51d43f058e2d		Head Nodes, Worker nodes D12 v2 (x2), D4 v2 (x4)		

Figura 1 - Cluster criado na Plataforma Azure.

Execuções:

N=1000

Target Cloud RTL --> Downloaded 3.0MB in 3s
Target Cloud RTL --> Decompressed 3.0MB in 0s
"contentType": OMPCLLOUD Runtime: 66.966480s
CPU Runtime: 10.234256s
Non-Matching CPU-GPU Outputs: 0
Libomptarget --> Unloading target library!
Target Cloud RTL --> Uploading = 9s
Target Cloud RTL --> Downloading = 3s
Target Cloud RTL --> Compression = 0s
Target Cloud RTL --> Decompression = 0s
Target Cloud RTL --> Execution = 53s

N=2000

Target Cloud RTL --> Downloaded 3.0MB in 3s
Target Cloud RTL --> Decompressed 3.0MB in 0s
"contentType": OMPCLLOUD Runtime: 66.966480s
CPU Runtime: 10.234256s
Non-Matching CPU-GPU Outputs: 0
Libomptarget --> Unloading target library!
Target Cloud RTL --> Uploading = 9s
Target Cloud RTL --> Downloading = 3s
Target Cloud RTL --> Compression = 0s
Target Cloud RTL --> Decompression = 0s
Target Cloud RTL --> Execution = 53s

N=3000

Target Cloud RTL --> Downloaded 3.0MB in 3s
Target Cloud RTL --> Decompressed 3.0MB in 0s
"contentType": OMPCLLOUD Runtime: 66.966480s
CPU Runtime: 10.234256s
Non-Matching CPU-GPU Outputs: 0
Libomptarget --> Unloading target library!
Target Cloud RTL --> Uploading = 9s
Target Cloud RTL --> Downloading = 3s
Target Cloud RTL --> Compression = 0s
Target Cloud RTL --> Decompression = 0s
Target Cloud RTL --> Execution = 53s

N=4000

Target Cloud RTL --> Downloaded 61.0MB in 11s
Target Cloud RTL --> Decompressed 61.0MB in 0s
"contentLanguage": nOMPCLLOUD Runtime:
229.108690s
CPU Runtime: 1114.728817s
Non-Matching CPU-GPU Outputs: 0
Libomptarget --> Unloading target library!
Target Cloud RTL --> Uploading = 27s
Target Cloud RTL --> Downloading = 11s
Target Cloud RTL --> Compression = 9s
Target Cloud RTL --> Decompression = 0s
Target Cloud RTL --> Execution = 196s

N=5000

Target Cloud RTL --> Downloaded 95.0MB in 16s
Target Cloud RTL --> Decompressed 95.0MB in 0s
"contentLanguage": n
OMPCLLOUD Runtime: 387.681192s
CPU Runtime: 2660.857864s
Non-Matching CPU-GPU Outputs: 0
Libomptarget --> Unloading target library!
Target Cloud RTL --> Uploading = 29s
Target Cloud RTL --> Downloading = 16s
Target Cloud RTL --> Compression = 15s
Target Cloud RTL --> Decompression = 0s
Target Cloud RTL --> Execution = 348s

N=8000 (RUN_TEST = 0)


Target Cloud RTL --> Downloaded 244.0MB in 31s
Target Cloud RTL --> Decompressed 244.0MB in 2s
"contentLanguage":OMPCLLOUD Runtime:
1300.378336s

Conclusões:

Para matrizes de tamanho de N grande, em que o tempo de execução do programa é alto, tornando o tempo total de transferência de dados entre o o Host e Clusters desprezível, verificou-se que executar na nuvem é uma boa opção, pois os recursos computacionais são maiores, havendo grandes ganhos de performace na execução do programa. Porém, para matrizes com tamanhos de N pequeno, o tempo total de upload dos dados do host para a Cloud, da execução, e do download dos resultados, acaba sendo maior do que a execução do programa na máquina local. Foi possível verificar também que o Spark Job do cluster direciona os trabalhos para executores, que então distribuem os trabalhos para os núcleos executores. Após a finalização dos trabalhos, ocorre então a etapa de coleta dos resultados feita também por esses núcleos.

Imagens

Processamentos realizados no Cluster referentes as valores de N: 1000, 2000, 3000, 4000 e 5000 e 8000.

2.0.2.2.5.4.2-7

History Server


Event log directory: wasb:///hdp/spark2-events

Show 20 entries

Search:

App ID	App Name	Attempt ID	Started	Completed	Duration	Spark User	Last Updated
application_1497621753722_0017	code-t12-parallel	1	2017-06-16 17:59:01	2017-06-16 18:00:29	1.5 min	sshuser	2017-06-16 18:00:31
application_1497621753722_0016	code-t12-parallel	1	2017-06-16 17:33:16	2017-06-16 17:34:47	1.5 min	sshuser	2017-06-16 17:34:47
application_1497621753722_0015	code-t12-parallel	1	2017-06-16 17:24:48	2017-06-16 17:25:40	52 s	sshuser	2017-06-16 17:25:40
application_1497621753722_0014	code-t12-parallel	1	2017-06-16 17:01:25	2017-06-16 17:02:11	46 s	sshuser	2017-06-16 17:02:11
application_1497621753722_0013	code-t12-parallel	1	2017-06-16 16:53:15	2017-06-16 16:55:37	2.4 min	sshuser	2017-06-16 16:55:37
application_1497621753722_0012	code-t12-parallel	1	2017-06-16 16:15:52	2017-06-16 16:18:19	2.5 min	sshuser	2017-06-16 16:18:20
application_1497621753722_0011	code-t12-parallel	1	2017-06-16 16:03:11	2017-06-16 16:04:21	1.2 min	sshuser	2017-06-16 16:04:21
application_1497621753722_0010	code-t12-parallel	1	2017-06-16 15:52:58	2017-06-16 15:54:06	1.1 min	sshuser	2017-06-16 15:54:06
application_1497621753722_0009	code-t12-parallel	1	2017-06-16 15:45:46	2017-06-16 15:46:33	47 s	sshuser	2017-06-16 15:46:34
application_1497621753722_0008	code-t12-parallel	1	2017-06-16 15:30:06	2017-06-16 15:30:19	14 s	sshuser	2017-06-16 15:30:20
application_1497621753722_0006	code-t12-parallel-3000	5	2017-06-16 15:04:14	2017-06-16 15:04:29	15 s	sshuser	2017-06-16 15:04:30
		4	2017-06-16 15:03:52	2017-06-16 15:04:10	19 s	sshuser	2017-06-16 15:04:11
		3	2017-06-16 15:03:31	2017-06-16 15:03:46	15 s	sshuser	2017-06-16 15:03:46
		2	2017-06-16 15:03:10	2017-06-16 15:03:24	14 s	sshuser	2017-06-16 15:03:25
		1	2017-06-16 15:02:52	2017-06-16 15:03:07	15 s	sshuser	2017-06-16 15:03:08

Job Exmplo para a execução N=3000

2.0.2.2.5.4.2-7

Jobs

Stages

Storage

Environment

Executors

SQL

code-t12-parallel application UI

Spark Jobs (?)

User: yarn

Total Uptime: 1.5 min

Scheduling Mode: FIFO

Completed Jobs: 2

Event Timeline

Completed Jobs (2)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	collect_at_kernel_spark.scala:93	2017/06/16 18:00:06	22 s	1/1	24/24
0	collect_at_kernel_spark.scala:72	2017/06/16 17:59:39	26 s	1/1	24/24

Details for Job 1

Status: SUCCEEDED
Completed Stages: 1

- ▶ Event Timeline
- ▶ DAG Visualization

Completed Stages (1)

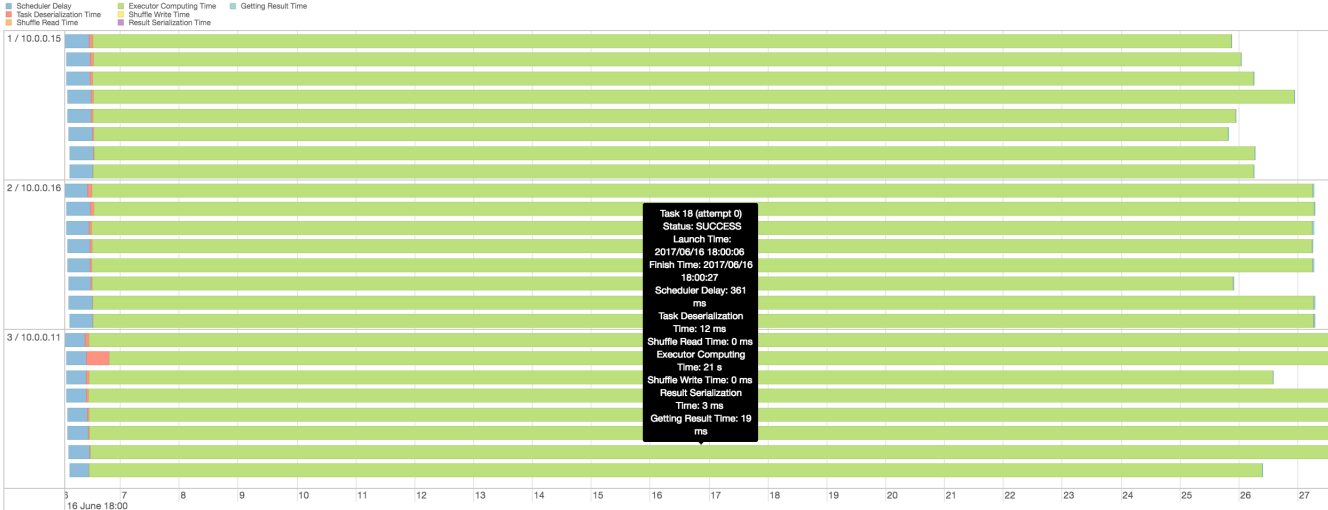
Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
1	collect at kernel.spark.scala:93 +details	2017/06/16 18:00:06	22 s	24/24				

Event Timeline

Details for Stage 1 (Attempt 0)

Total Time Across All Tasks: 8.1 min
Locality Level Summary: Process local: 24

- ▶ DAG Visualization
- ▶ Show Additional Metrics
- ▶ Event Timeline
- ▶ Enable zooming



Summary Metrics for 24 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	19 s	20 s	21 s	21 s	21 s
GC Time	0 ms	14 ms	42 ms	63 ms	63 ms

Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Succeeded Tasks
1	10.0.0.15:42192	2.7 min	8	0	8
2	10.0.0.16:45765	2.8 min	8	0	8
3	10.0.0.11:43002	2.8 min	8	0	8

Tasks (24)

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Errors
0	24	0	SUCCESS	PROCESS_LOCAL	3 / 10.0.0.11	2017/06/16 18:00:06	21 s	63 ms	

Metric Actions ▼

Last 1 hour ▼

HDFS Disk Usage

DataNodes Live

HDFS Links

Active NameNode

Standby NameNode

4 DataNodes

More... ▼

Memory Usage

Network Usage

CPU Usage

Cluster Load

NameNode Heap

NameNode RPC

0.20 ms

NameNode CPU WIO

NameNode Uptime

8.2 hr

ResourceManager Heap

ResourceManager Uptime

8.2 hr

YARN Memory

NodeManagers Live

4/4