View metadata, citation and similar papers at core.ac.uk
brought to you by CORE

Raspberry Piを使用したSCの制作

システム科学技術学部　知能メカトロニクス学科, 情報工学科
1年　鈴木　キリロ
1年　茂木　星南
指導教員　システム科学技術学部　知能メカトロニクス学科
准教授　間所　洋和
教授　佐藤　和人
指導補助　システム科学技術学部 機械知能システム学科
4年　橋本 真澄
4年　渡邊 拓磨

1. 研究目的

本研究では，コンピュータクラスタ(以下では，単にクラスタと表記)と人工知能の仕組みの理解を目的とする．また，低コストで多スレッドのクラスタ構築を目的とする．

2. 使用物品

本研究では，表 1 に示す物品を用いてクラスタを構築した．

表 1　本研究にて使用した物品

品名	型番	数量
Raspberry Pi 3 Model B	RS コンポーネンツ RS-122-5826	8 個
5V 出力 AC アダプター 6 ポート	omars AC13-02	2 個
Raspberry Pi 7" Touchscreen Display	ELEMENT14-2473872	2 個
16 ポートギガビットスイッチングハブ	tp-link TL-SG1016D	1 個
メタル TV ラック	不二貿易 MTV-2CR	1 個
PowerLine Micro USB	ANKER B81232016	3 セット
6ACat LAN ケーブル	Buffalo BSLS6AFU05BK	10 個
5V 30mm ファン	ICFAN F3020AP-05MCW	3 個
ナイロンスタンドオフスペーサー　100 個	uxcell a14051300ux0024	1 セット
スペーサ 50 個	廣杉計器 BSB-2620E	1 セット
ケーブルタイ 2.5mm×99mm　100 本	TRUSCO TRJ100B	1 セット
4 個口　埃シャッター付き電源タップ 3m	ELECOM T-ST02-22430WH	1 個
microSDHC UHS-I CARD 16GB	TOSHIBA THN-M301R0160A4	10 個
モバイルルータ	ROMIX　4GUFI906	1 個

3. 研究内容

本研究では，Raspberry Pi でクラスタを構成し，spark・hadoop を使用した並列処理や分散処理

を行った．さらに外部からクラスタを操作できる仕組みを構築した．

3.1. Raspberry Pi を用いたクラスタの構成

Raspberry Pi を 8 台使用し，それらを繋げてクラスタを構成した．8 台のうち 1 台をマスターとし，8 台すべてに hostname とその Raspberry Pi の IP アドレスを記載した．

マスターには master から slave へ SSH する際，パスワードの入力を求められないように鍵を配置した．これらの Raspberry Pi と USB 充電器，スイッチングハブを TV ラックに積載固定し，クラスタとして集積した．

構築したクラスタを図 1 に示す．

図 1　構築したクラスタの外観

左上が 8 台の Raspberry Pi，右下がスイッチングハブである．

3.2. spark を使用した処理例

並列分散処理のフレームワークとして，本研究では PySpark を用いた．開発環境となる jupyter から PySpark を起動し，spark 上でコードを実行できるようにする．図 2 にコードと出力結果を示す．

```
In [1]: import os, sys
from datetime import datetime as dt
print ("loading PySpark setting...")
spark_home = os.environ.get('SPARK_HOME', None)
if not spark_home:
    raise ValueError('SPARK_HOME environment variable is not set')
sys.path.insert(0, os.path.join(spark_home, 'python'))
sys.path.insert(0, os.path.join(spark_home, 'python/lib/py4j-0.8.2.1-src.zip'))
with open(os.path.join(spark_home, 'python/pyspark/shell.py')) as f:
    code=f.read()
    exec(code)

loading PySpark setting...
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0
      /_/

Using Python version 3.5.3 (default, Sep 27 2018 17:25:39)
SparkSession available as 'spark'.
```

図 2　PySpark の起動

データセットを読み込み，散布図として表示し，クラスタリングにより 3 種類に色分けをする．データセットには，特徴量として“setosa”，“versicolor”，“virginica”アイリス(アヤメ)のがく片・花弁の幅及び長さが含まれている．プログラムでは，最初にがく弁・花弁の長さのみを抜き出して表示する．X ががく片の長さ，Y が花弁の長さである．

なお，赤色の点が“setosa”，緑色の点が“versicolor”，青色の点が“virginica”のデータである．図 3 にコードと出力結果を示す．

```
In [2]: %matplotlib inline
import matplotlib.pyplot as plt
import datetime
import numpy as np
import pandas as pd
from sklearn import datasets
from pyspark.mllib.clustering import KMeans
plt.style.use('ggplot')

iris = datasets.load_iris()
plt.figure(figsize=(9,7))

for i, color in enumerate('rgb'):
    idx = np.where(iris.target == i)[0]
    plt.scatter(iris.data[idx,0],iris.data[idx,2], c=color, s=30, alpha=.7)

plt.show()
```

図 3　がく片・花弁の幅及び長さのデータ

続いて，教師なしクラスタリング手法の一つ K-Means を使用し，3 種類の花の中心を表示するように設定する．機械学習ライブラリの“mllib”と K-Means を呼び出し，data=sa.parallelize でデータを変換し，分散処理が実行できるように変更している．図 4 にコードと出力結果を示す．

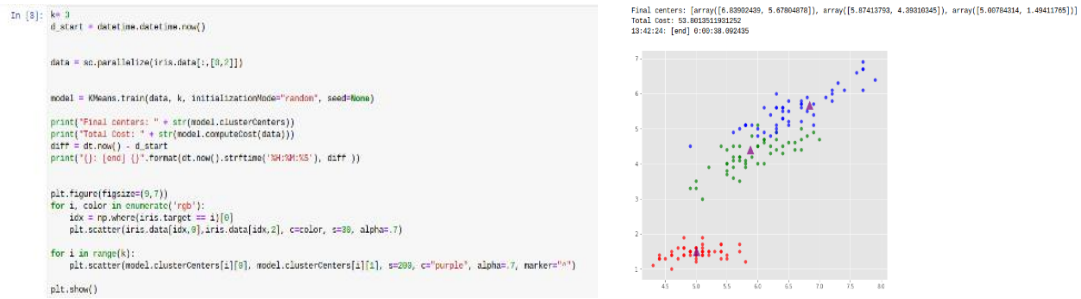


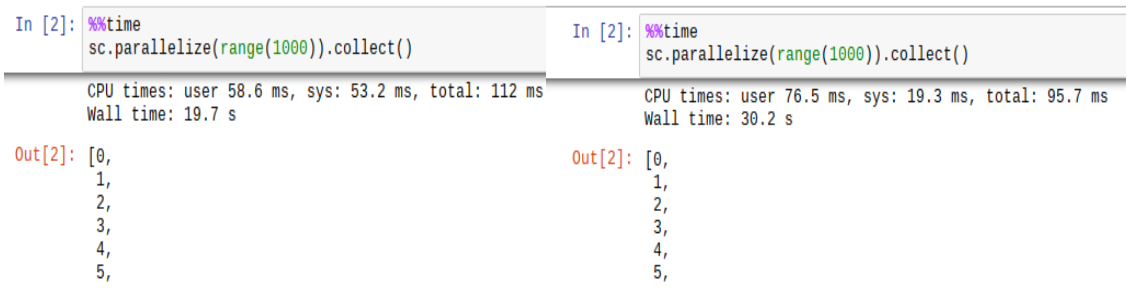
図 4　分散処理の例

図 4 の学習データから，どの特徴点がどこに分類されるのかを分析する．図 5 にコードと出力結果を示す．

図 5 各データの範囲

図 6 では処理する台数を変更している．図 6(a)では 6 台，図 6(b)では 1 台で処理させている．0 から 1000 までを読み込むという簡単な処理だがコアとメモリが共有されるため，完了するまでに 10 秒程度の時間差が出ている．

1 台ではメモリ 512MB 4 コアが処理に使われているのに対し，7 台では 3GB　24 コアを使用していた．処理の結果と計算所要時間を図 6 に示す．



(a)6 台で処理した場合　　(b)1 台で処理した場合

図 6　処理台数を変更した場合の計算所要時間の比較

3.3. 外部からのクラスタの操作

取得した SIM の IP アドレスを使用して SSH や VNC でマスターを外部ネットワークから操作をできるようにした．マスター経由で他の Raspberry Pi も操作できるため外部からすべてのノードを操作できるようになり，その結果遠隔操作が実現し，研究室外での研究時間を増やすことができた．図 7 は本研究にて使用したモバイルルータである．

(a)未使用時　　(b)使用時

図 7　使用したモバイルルータ

4. 研究から得られた成果

- Raspberry Pi でクラスタを構成し，hadoop や spark を用いた並列処理や分散処理を行うことができた．
- 並列処理や分散処理を hadoop や spark の HDFS や RDD という仕組みを理解できた．
- 実践的なプログラミングによって，python やスクリプトの技術が向上した．
- プログラミングを通じて，アルゴリズムに関する知識が身についた．
- Linux に関する知識が身につき，早く正確に操作できるようになった．
- SSH や VNC 等の通信プロトコルに関する知識を深めることができた．

5. まとめ

今回の研究で, spark を用いた分散処理や並列処理について学ぶことができた. プログラムがどのようにデータを扱うのかを少しずつ理解していくことができ，データの分析をすることができた．しかし，分散処理は非常に難解であった．切り口として，分散処理の場合には Hadoop・Spark の両方を使う方法を試みた．しかしながら，完全動作には至らなかった．機械学習によるデータ分析は実現できたが，クラスタの仕組みは完全には理解できなかった．

研究を効率よく進めようと試みるうえで，通信や Linux に関する知識が深まり，外部ネットワークからの操作が可能になった．データの解析をする際に python のライブラリである pandas や mathplotlib などを用いる必要があり，コンピュータを用いて数学的な解析をするための知識が身に付いた．また，不具合の原因の究明やセキュリティなどの課題も見つかった．今回達成できなかった分散処理の理解と実行は，次回の研究における課題となる．

〈参考文献〉

[1]Apache Spark 入門　動かして学ぶ最新並列分散処理フレームワーク 株式会社 NTT データ
[2]Apache Spark： https://spark.apache.org/