

Pràctica Scala

Enric Rodriguez, Marc Cané

29 de novembre de 2018

Índex

1	Exemple de titol	3
2	Titol 2	4
2.1	Per Coordenades	4
2.2	Per files	4
2.2.1	Exemple	4
2.2.2	Implementació del mètode CSR	5
2.3	Codi	6

Capítol 1

Exemple de titol

Quan parlem de matrius disperses ens referim a matrius de grans dimensions en la qual la majoria d'elements son zero. Direm que una matriu és dispersa, quan hi hagi benefici en aplicar els mètodes propis d'aquestes.

Per identificar si una matriu és dispersa, podem usar el següent:

Una matriu $n \times n$ serà dispersa si el número de coeficients no nuls es $n^{\gamma+1}$, on $\gamma < 1$.

En funció del problema, decidim el valor del paràmetre γ . Aquí hi ha els valors típics de γ :

- $\gamma = 0.2$ per problemes d'anàlisi de sistemes elèctrics de generació i de transport d'energia.
- $\gamma = 0.5$ per matrius en bandes associades a problemes d'anàlisi d'estructures.

Capítol 2

Titol 2

2.1 Per Coordenades

És la primera aproximació que podríem pensar i és bastant intuïtiva. Per cada element no nul guardem una tupla amb el valor i les seves coordenades: (a_{ij}, i, j) .

A la realitat però, aquest mètode d'emmagatzemar les dades és poc eficient quan hem de fer operacions amb les matrius.

2.2 Per files

També conegut com a *Compressed Sparse Rows (CSR)*, *Compressed Row Storage (CRS)*, o format *Yale*. És el mètode més estès.

Consisteix en guardar els elements ordenats per files, guardar la columna on es troben, i la posició del primer element de cada fila en el vector de valors. Així ens quedaran tres vectors:

- **valors:** de mida n_z , conté tots els valors diferents.
- **columnnes:** també de mida n_z , conté la columna on es troba cada un dels elements anteriors.
- **iniFiles:** de mida $m + 1$, conté la posició on comença cada fila en els vectors valors i columnnes, sent m el nombre de files de la matriu.

2.2.1 Exemple

Si es canvien files per columnnes, dona la implementació per columnnes, o també anomenada *Compressed Sparse Columns (CSC)*.

2.2.2 Implementació del mètode CSR

Hem implementat un script Matlab amb una classe `CSRSparsedMatrix` que guardi les dades necessàries. Aquestes les tenim en “l’atribut” `Matrix` dins del bloc `properties` (línia 10 del codi següent). Aquestes dades consisteixen en el següent:

- `Matrix.nColumns`: número de columnes de la matriu, necessari per recrear les files posteriorment.
- `Matrix.values`: vector valors comentat anteriorment, amb els valors no nuls de la matriu.
- `Matrix.columns`: vector de columnes, amb la columna corresponent a cada valor amb el mateix índex.
- `Matrix.beginningRow`: vector amb els índex comença cada fila en el vector de valors i de columnes.

2.3 Codi

Exemple codi Scala:

```
object MapReduceEnric{
  def mapping1(file_name: String, file: (String, List[String])): List[(String, (String,
    Int))] = {
    val wordList = FirstHalf.readFile(file._1).split("
    +").toList.filterNot(file._2.contains(_))

    val x = (for(word <- wordList) yield (file_name, (word, 1)))//.groupBy(_._1)
    x
  }

  //key-> Filename, values-> list of (Word, count)
  def reducing1(key: String, values: List[(String, Int)]): List[(String, Int)] = {
    val res = for( (word, count_list) <- values.groupBy(_._1).toList )
      //For every pair of word and list of counts, add up its counts
      yield (word, count_list.map( {case (_, count) => count } ).reduceLeft( _ + _))

    res.sortWith(FirstHalf.moreFrequent)
  }

  def main1() = {
    val stopwords = FirstHalf.readFile("test/english-stop.txt").split(" +").toList
    val files = Main.openPgTxtFiles("test", "pg", ".txt")

    val input = ( for( file <- files) yield (file.getName, (file.getAbsolutePath,
      stopwords)) ).toList

    val system = ActorSystem("TextAnalyzer2")

    val master = system.actorOf(Props(new MapReduceActor[String, (String, List[String]),
      String, (String, Int)](input, MapReduceEnric.mapping1, MapReduceEnric.reducing1, 2,
      2)))
    implicit val timeout = Timeout(10 days)
    val futureResponse = master ? "start"
    val result = Await.result(futureResponse, timeout.duration)
    system.shutdown
    print(result)
  }
}
```
