

I. Research Overview and Outcome

Object Similarity MapReduce Template



- Given:
 - An object α of type γ
 - A collection β of n datasets β_1, \dots, β_n
 - Each β_i contains a list of objects $\beta_{i,1}, \dots, \beta_{i,m_i}$ of type γ
 - A similarity function $\delta(\alpha, \alpha')$ that determines the similarity between α and α' of type γ
- Find the level of similarity between α and every object in β using δ .
- Output the results organized by key based on, for example, level of similarity.
 - i.e. {0-30% Similar, γ }[Objects], {31-60% Similar, γ }[Objects], (etc)

Some Possible Applications

- Social Network Profiles**
 - Given a profile α and its corresponding friend list F_α
 - Return all friends that have an 80% similarity based on musical taste.
- Spatial Objects**
 - Given a search point α and an unsorted dataset of all McDonalds restaurants
 - Return all McDonalds within 6-10 miles of α that have a "Play Place"
- Plagiarism Detection**
 - Given a document α and a corpus of documents β
 - Find if there is a member β_i of β with a level of similarity to α greater than 50%

Traditional Implementation

- for(int i = 0; i < n; i++)
 - Read dataset β_i
 - for(int j = 0; j < β_i .numRecords; j++)
 - Read object $\beta_{i,j}$
 - similarityMeasure = $\delta(\alpha, \beta_{i,j})$
 - Place $\beta_{i,j}$ in appropriate "bucket" based on similarityMeasure
- Sequential
- Bottlenecks to one object comparison at a time
- Data is in one central location
- Requires a write to file each time an additional result is obtained



Why the need to cope with heterogeneous datasets?

- The availability of linked open data (LOD) and other heterogeneous data allows us to dynamically add functionality to our applications
- Our GPS navigation system may not have a list of all McDonalds restaurants but, if we tell it where to find a information then a method such as IBM's Midas can help us extract the data and we can instantly provide new functionality.
- MapReduce allows us to focus less on efficiency and more on dreaming up new content avenues we can open
- Once we see a consistent need for the functionality we can proceed to develop an efficient non-mapReduce solution.
- MapReduce allows us to provide the functionality right away without significant effort and development

Unsorted Spatial Data Example

- Given a point α , a set of keywords ϵ , and a collection of datasets β
- Return a list of objects in β that contain some or all ϵ , aggregated by "distance range".
 - Given:
 - α = (80.98, -127.356)
 - ϵ = {McDonalds}
 - Result:
 - List of
 - All McDonalds between 0-5 miles of α
 - All McDonalds between 6-10 miles of α
 - All McDonalds between 11-20 miles of α
 - All McDonalds 21 miles or more away from α
- Paradigm can be tuned quite easily for many different domains!

MapReduce Implementation

- Input to Mapper $\langle \beta_i, \beta_{i,j} \rangle$
 - similarityMeasure = $\delta(\alpha, \beta_{i,j})$
 - $k =$ whichBucket(similarityMeasure)
 - outputIntermittentKeyValue($k, \beta_{i,j}$)
- Intermediate Output \langle key k , Object $\beta_{i,j}$ \rangle
- Combiner will combine these by key
- Input into Reducer \langle key k , $\langle \gamma \rangle$ ObjectIterator \rangle
- Reducer Output \langle key k , γ [] Object \rangle
 - In the case of the described spatial application the result is simple a concatenated file of all of the objects in the ObjectIterator
 - The result can be in any format we specify; array, iterator, string, etc.

U.S. DEPARTMENT OF DEFENSE

A DoD Publications Archive Search Tool Using MapReduce

Data

- Approximately 4,000 HTML files in each of the 4 categories.
 - New files are added daily
- Size ranges from 34k to 500k
- Data was duplicated to reach Hadoop file quantity limit.
- Mapper will process entire file as record
 - Attempt will be made to adapt code to read one line at a time as well and output aggregate counts.
 - i.e. How many speeches were made on July 4, 1996?

Result

- A searchable data structure containing information about all DoD publications since 1994.
- Query front end can easily be developed
- Information discovery and data mining possibilities will exist.

Overview

MapReduce Paradigm

- Each file is processed as a record $\langle k1, v1 \rangle$
 - \langle NullWritable, BytesWritable fileContents \rangle
- Who, What, When is extracted and emitted $\langle k2, v2 \rangle$
 - \langle Text Who, Text When \rangle
 - \langle Text When, Text What \rangle
 - \langle Text When, Text Who \rangle
- Three Possibilities for Reduction:
 - Reducer aggregates based on Who and returns a list of Whens
 - \langle Text Who, BytesWritable Whens \rangle
 - Reducer aggregates based on When and returns a list of Whats
 - \langle Text When, BytesWritable Whats \rangle
 - Reducer aggregates based on When and returns a list of Whos
 - \langle Text When, BytesWritable Whos \rangle

Sample Application

- Find all speeches and transcripts of President Bush in 1995
- Find any press advisories about Saudi Arabia that were issued at most 5 days before a speech given by Donald Rumsfeld that mentions Saudi Arabia.

II. PIRE Experiences



- Received guidance from leading researchers on:
 - IBM's systemT, JAQL, and Midas
 - MapReduce using Hadoop
 - JSON, DB2, and more!
- Immersed myself in Bay Area and Catalonian culture.
- Used the knowledge gained to propose a data driven outcome prediction system.
- Extracted the US government organizational chart from PDF files using systemT, JAQL, JSON, and Hadoop!
- Gained intimate knowledge of US government agencies encouraging my pursuit of civil service careers.
- Made friends from Silicon Valley, China, Spain, Italy, and more!



III. Acknowledgement

The material presented in this poster is based upon the work supported by the National Science Foundation under Grant No. OISE-0730065, IIS-0837716, CNS-0821345, HRD-0833093, and IIP-0829576. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.