

PROJECTS

Deadline: 3rd, May 2015 (23:55 Dublin Time)

IMPORTANT NOTICE: STUDENTS ONLY USE FREE TOOLS/SERVICES. WE WILL NOT BE LIABLE FOR ANY REFUND, COMPENSATION OR ANY COSTS INCURRED BY THE STUDENTS.

Submission: Submission should take place via moodle on or before the deadline. Submission should consist of one zip file (ClassNo_Surname_FirstName_StudentNo_Project.zip) which contains:

1. Source codes
2. Testing data
3. Report (~10 pages) in PDF

Students can choose one of the following projects:

Project- 1 (AWS-SimpleDB)	1-Student
<p>Amazon SimpleDB is a scalable, flexible and non-relational data store that enables you to store and query data items using web services requests. In order to access data, SimpleDB uses Amazon SimpleDB scratchpad, which is a web-based user interface. Despite this scratchpad is simple, it is not user friendly. In this project, using AWS SDK, the students will develop a SimpleDB management tool. This tool has a graphic interface and offers the following main functions (operations):</p> <ol style="list-style-type: none">a) Create domainb) Add/Update/Delete Items in domainc) Query domaind) Delete domaine) List domain <p>Students can add more operations. Students can refer to some SimpleDB third party management tools such as SDB tool plugin, Mindscape, etc.</p>	

Project- 2 (Online Calendar)	1-2 student(s)
<p>Online calendar such as Google Calendar is a time-management web application. It allows users to create new events, edit or delete existing events, and search for events. Indeed, it also allows users to share their calendars. In this project, the students are asked to develop an online calendar with the following operations:</p> <ul style="list-style-type: none"> a) Create new events, b) Edit or delete existing events, c) Search for events, d) Time slots booking, e) Share calendar. <p>This online calendar can be stored in cloud-based storage. Students can use any web development tools to develop this online calendar.</p>	

Project- 3 (Hadoop/MapReduce-log analysis)	1-Student
<p>In this project, you will gain hands-on experience with the MapReduce programming model building and running an application using Hadoop. A common task in digital forensics is the analysis of log files. Each line (entry) of a web server's log file normally contains important information such as: IP address, data and time of request, request line, HTTP status, etc.</p> <p>Write a MapReduce program that takes a log file as an input and then extracts the following information from the log files:</p> <ul style="list-style-type: none"> a) Total number of connections to the server (i.e. total numbers of entries). b) List of distinct IPs. c) Number of entries for each IP. d) List of distinct IPs, number of entries for each IP in a given periods (e.g. from 01/02/2015 to 28/02/2015) <p>You can use any Hadoop platforms to test your program (AWS MapReduce/Hadoop, Google Hadoop Cloud, etc.)</p>	

Project- 4 (Hadoop/MapReduce-Link analysis)	1-Student
<p>In this project, you will gain hands-on experience with the MapReduce programming model by designing and implementing an application using Hadoop. Link analysis techniques are applied to data that can be represented as nodes and links. Link analysis can be used in social network analysis to answer the following questions:</p> <ul style="list-style-type: none"> (i) Which are the powerful users? (ii) Which users influence other members? (iii) Who is relatively isolated, and who is well connected? etc. <p>Link analysis can also be applied in search engines to evaluate important documents. In this project, students are asked to: Get a list of nodes and links as input (a node represents an entity such as a person; a link represents a relationship between two entities such as a relationship between two people). A list of nodes and links is also called a graph. Write a MapReduce program that:</p> <ul style="list-style-type: none"> a) Computes degree, closeness and betweenness of each node b) Finds cutpoints, K-plexes and K-cores of the input graph c) Finds nodes that are structure equivalence. <p>(The definition of degree, closeness, betweenness, cutpoints, K-plexes, Kcores and structure equivalence is attached). You can use any Hadoop platforms to test your program (AWS MapReduce/Hadoop, Google Hadoop Cloud, etc.)</p>	

Project- 5 (Hadoop/MapReduce-Text file analysis)	1-Student																												
<p>In this project, you will gain hands-on experience with the MapReduce programming model building and running an application using Hadoop. Today, we have a deluge of text data from various sources, such as e-papers, research papers, books, digital libraries, email and Web pages. Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Users need new techniques to analysis text data such as comparing different documents, ranking the importance and relevance of the documents, or finding patterns across multiple documents. In this project, student implements a simple technique of document clustering analysis by using MapReduce programming model. This program consists of the following functions:</p> <ul style="list-style-type: none">a. Represent input documents as a table of term vectors.b. Reduce document space (i.e. reduce the number of dimensions)c. Implement K-Means algorithm to cluster input documents <p>You can use any Hadoop platforms to test your program (AWS MapReduce/Hadoop, Google Hadoop Cloud, etc.)</p> <p>An example of a table of term vectors:</p> <table><tr><th></th><th>an</th><th>apple</th><th>is</th><th>not</th><th>orange</th><th>the</th></tr><tr><th>Doc1</th><td>1</td><td>5</td><td>8</td><td>22</td><td>0</td><td>32</td></tr><tr><th>Doc2</th><td>0</td><td>6</td><td>10</td><td>19</td><td>0</td><td>13</td></tr><tr><th>Doc3</th><td>3</td><td>12</td><td>15</td><td>4</td><td>8</td><td>5</td></tr></table> <p>The number in each cell is the number of occurrences of a term in a relevant document.</p>			an	apple	is	not	orange	the	Doc1	1	5	8	22	0	32	Doc2	0	6	10	19	0	13	Doc3	3	12	15	4	8	5
	an	apple	is	not	orange	the																							
Doc1	1	5	8	22	0	32																							
Doc2	0	6	10	19	0	13																							
Doc3	3	12	15	4	8	5																							

Project- 6 (Facebook like application)	2-3 Student
<p>In this project, students should build a small Facebook-like application by using any web development tools and cloud-based database storage (such as Amazon's SimpleDB).</p>	

Submission: Submission should take place via moodle on or before the deadline.

Submission should consist of one zip file

(ClassNo_Surname_FirstName_StudentNo_Project.zip) which contains:

- 1. Source codes**
- 2. Testing data**
- 3. Report (~10 pages) in PDF**