# Distributed Systems Assignment

Release Date: 10th October
**Assignment Deadline: 17th November (4 pm)**
Feedback Return: 1st December

This assignment is worth 25% of the final mark. The assignment is a distributed programming exercise using the Apache Ignite framework and will be marked out of 25 (=100%).

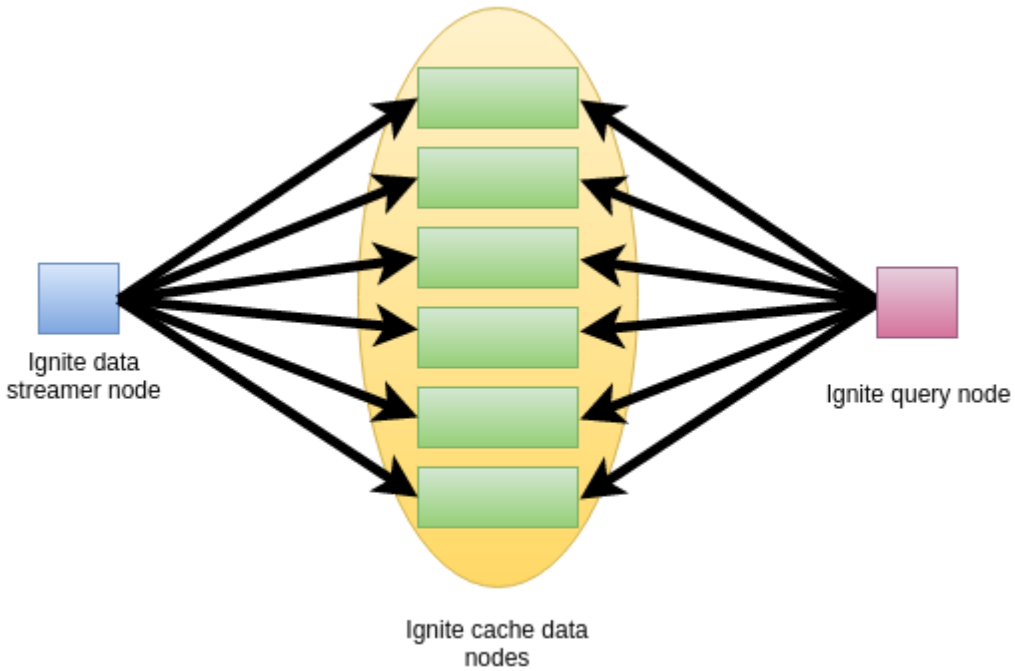
Please read the full description of the assignment before starting to solve.

## Assignment Description

This assignment requires you to implement a distributed system using the Apache Ignite framework to get top ten most popular Wikipedia pages. You are given an hourly page view log of accesses Wikipedia pages, i.e. which page is viewed how many times in that hour: https://dumps.wikimedia.org/other/

### System Description

Your implementation should include, at least, a streamer node which streams Wikipedia log data to the Ignite cache data nodes, and a query node which queries the data nodes for top 10 most popular pages (Fig 1) every 10 seconds.



Data is to be concurrently processed by all the Ignite cache data nodes. The data cache is to be configured to operate using a sliding window of one second. The query node should be able to query the streaming data continuously.

Input Data file format

WIkipedia page view logs are plain text files. Each file comprises usage statistics for a time period of one hour. Each line in a log file has four fields, separated by space:

- Field 1: denotes if the page is from wikibooks/wikidictionary/wikimedia… etc.
- Field 2: title of the page
- Field 3: Number of views of the page in that hour
- Field 4: Size of the content returned

Example line from a log:

fr.b Special:Recherche/All_Mixed_Up 1 730

Refer https://dumps.wikimedia.org/other/pagecounts-raw/ for more detail about log files.

**Assignment Questions**

The assignment is divided into three parts.

Part A (marks: 10/25):

1. Create cache configuration with sliding window of **1 second**.
2. Create a ignite streamer node to read from a log file and stream the data to ignite data cache nodes.
3. Create a query node to continuously query every **10 seconds** for top 10 most popular pages in the log. Output of the query should be as per the output format specified below.

Make sure to experiment with multiple data nodes.

Part B (marks: 10/25):

1. Create three ignite streamer nodes to stream content from three different log files simultaneously. Your application should take the log file path as a command line argument.
2. Run all the streamers to concurrently feed data to the data cache nodes.
3. Run the query node created in part A to continuously retrieve top 10 most popular pages among three log files being processed. Output of the query should be as per the output format specified below.

Part C (marks: 5/25):

1. Modify the program created in part B to determine the top ten most popular pages viewed from Wikibooks as specified by the three page count log files. Note that each page view statistics from Wikibooks has ".b" in its first column in the log file.
2. Run the streamers and the query nodes as part B. Output of the query should be as per the output format specified below.

Output format:

Query node should log the statistics of top ten popular pages in **descending order** of their popularity for all of the queries it makes in a single execution run. Each page statistics should be in a separate line and should follow below format:

time_when_query_result_is_received:visit_count:page_title

- time_when_query_result_is_received: Posix formed system time when query result is received. Note that, this value will be same for all the 10 entries in a round, but different for subsequent rounds of queries.
- View_count: number of views of the page counted till the time the query is made.
- Page_title: contains the page title mentioned in the second column of the log file.

The output should be printed to console and as well as to "log-partX.txt" saved in the designated directory. There should be separate log files for the three parts of the assignment, and "X" in the filename "log-partX.txt" should be "A" / "B" / "C" corresponding to the part number.

## Submission Instructions

Your submission should be organized into the following directory structure:

```
<your_uun>
    |---------- readme.txt - any specific design decision you want us to know
    |---------- source      - contains all source files (ideally your idea project directory)
    |---------- build       - compiled binary (jar file) (optional)
    |---------- dependency - contain all jars required for compilation except standard java jar files.
    |---------- run.sh
    |---------- log
                  |------ log-partA.txt
                  |------ log-partB.txt
                  |------ log-partC.txt
```

run.sh: The file should contain script with following options:

- ./run.sh compile : compiles the source and builds binary into corresponding directory.
- ./run.sh partA <log_filepath> : runs part A with log file path as an input

- ./run.sh partB <log_filepath1> <log_filepath2> <log_filepath3> : runs part B with 3 log files
- ./run.sh partC <log_filepath1> <log_filepath2> <log_filepath3> : runs part C with 3 log files

Note that, Your code should compile and run in DICE environment. You need to copy all required dependent jar files into the dependency directory for successful compilation and running. Note, in run.sh if you use maven for compilation, then you don't need to copy the jars.

You should run your code with logs taken from the Wikipedia page usage dataset https://dumps.wikimedia.org/other/pagecounts-raw/ and save the logs in corresponding files in log directory. You should also mention the download link of the log files used in the experiments in the readme file.

Make a tar.gz file of the project folder named as <your_uun>.tar.gz and submit using following command from your DICE machine:

submit ds 1 ds_assignment_<your uun>

Related resources:

1. https://dzone.com/articles/apache-ignite-word-count
2. https://ignite.apache.org/features/streaming.html

**University regulations**

On good Scholarly Practice. Please remember the University requirement as regards all assessed work. Details about this can be found at: http://web.inf.ed.ac.uk/infweb/admin/policies/academicmisconduct and at http://www.inf.ed.ac.uk/admin/ITO/DivisionalGuidelinesPlagiarism.html.

Remember, if you use ideas from elsewhere (including other students), cite them. And try not use too much of these. The regulation says you can pick up "general ideas" but not "pivotal ideas". But "general" and "pivotal" are very subjective and depends very much on the person making the judgement. Play safe and avoid getting into trouble.