POD
Trabajo Práctico Especial

Para el siguiente proceso se van a procesar la información de peliculas obtenidas de imdb.com (obtenidas a través de la api http://www.omdbapi.com/).
La información se proporciona en archivos que contienen un array de objetos JSON donde cada elemento tiene el siguiente formato (el campo Type determina si es una película o una serie y en todos los casos se indica con el valor “N/A” si no hay valor para dicho campo):


```
{
    "Title": "The Dark Knight Rises",
    "Year": "2012",
    "Rated": "PG-13",
    "Released": "20 Jul 2012",
    "Runtime": "165 min",
    "Genre": "Action, Thriller",
    "Director": "Christopher Nolan",
    "Writer": "Jonathan Nolan (screenplay), Christopher Nolan (screenplay), Christopher
Nolan (story), David S. Goyer (story), Bob Kane (characters)",
    "Actors": "Christian Bale, Gary Oldman, Tom Hardy, Joseph Gordon-Levitt",
    "Plot": "Eight years after the Joker's reign of anarchy, the Dark Knight is forced
to return from his imposed exile to save Gotham City from the brutal guerrilla terrorist
Bane with the help of the enigmatic Selina.",
    "Language": "English",
    "Country": "USA, UK",
    "Awards": "Nominated for 1 BAFTA Film Award. Another 42 wins & 88 nominations.",
    "Poster":
"http://ia.media-imdb.com/images/M/MV5BMTk4ODQzNDY3Ml5BMl5BanBnXkFtZTcwODA0NTM4Nw@@._V1_
SX300.jpg",
    "Metascore": "78",
    "imdbRating": "8.5",
    "imdbVotes": "1,030,777",
    "imdbID": "tt1345836",
    "Type": "movie",
    "tomatoMeter": "87",
    "tomatoImage": "certified",
    "tomatoRating": "8.0",
    "tomatoReviews": "327",
    "tomatoFresh": "283",
    "tomatoRotten": "44",
    "tomatoConsensus": "The Dark Knight Rises is an ambitious, thoughtful, and potent
action film that concludes Christopher Nolan's franchise in spectacular fashion.",
    "tomatoUserMeter": "90",
    "tomatoUserRating": "4.3",
    "tomatoUserReviews": "1203836",
    "DVD": "03 Dec 2012",
    "BoxOffice": "$448.1M",
    "Production": "Warner Bros. Pictures",
    "Website": "http://www.thedarkknightrises.com/",
    "Response": "True"
}
```

Se requiere generar una aplicación de consola que realice las siguientes queries sobre un grupo de películas/series:

1. Los **N** actores (de películas) más populares (popularidad se da por la cantidad de votos que recibieron en IMDB). Donde **N** lo provee el usuario.
2. Por cada año, mayor al año **Tope**, las películas más aclamadas por la crítica (todas las que tienen el valor mayor de Metascore). Donde **Tope** lo provee el usuario.
3. Las parejas de actores que más veces actuaron juntos y para cada una de ellas cuáles fueron las películas en las que actuaron.
4. Por cada director cuál es su actor (o actores) fetiche, o sea los que actuaron en más películas del director.

Cada corrida de la aplicación realiza una de estas queries sobre los datos obtenidos a partir de un archivo de texto..
La información de cuál es la query a correr, el del path al archivo con los datos y otros valores que se requieran para la query especifica (Tope o N), se recibe a través de argumentos de línea de comando al llamar a la aplicación.

Ejemplo: $> java query=2 tope=1996 path=/tmp/movies.json edu.itba.pod.hazel.tp.Main Corre la query 2, películas aclamadas por la crítica para años superiores a 1996, sobre las películas en el archivo /tmp/movies.json

La aplicación debe entonces correr la query y su salida (impresa en pantalla) debe ser la respuesta a la query. Además de la respuesta y para medir performance se pide que durante la ejecución se guarden los timestamp de los siguientes momentos:

- Inicio de la lectura del archivo.
- Fin de lectura del archivo.
- Inicio del trabajo map/reduce.
- Fin del trabajo map/reduce (incluye la impresión de las respuestas).

Todos estos momentos deben ser impresos en la salida luego de la respuesta con el timestamp en formato: dd/mm/yyyy hh:mm:ss:xxxx, deben ser claramente identificables.

Condiciones del trabajo práctico.

- El trabajo práctico debe realizarse en grupos de a dos personas exclusivamente.
- Cada una de las opciones debe ser implementada como un job map/reduce que pueda correr en un ambiente distribuido utilizando un grid de hazelcast.
- Los componentes del job, clases del modelo, test y el diseño de cada elemento del proyecto queda a criterio del equipo, pero debe estar enfocado en:
    - Que funcione correctamente en un ambiente concurrente, map/reduce en hazelcast.

    - Que sea eficiente para un gran volumen de datos.
- En campus.itba.edu.ar en la carpeta **Trabajo Práctico Especial** se iran dejando archivos con la información de peliculas/series con diferentes cantidades de elementos para poder pobar con diversos volúmenes de datos. Todos estos archivos deberán ser incluidos en el proyecto que se entrega y utilizados para probar.

Se debe entregar

- El código fuente de la aplicación.
- Un documento explicado:
    - Como preparar el entorno a partir del código fuente para ejecutar la aplicación en un ambiente con varios nodos.
    - Brevemente como se diseño los componentes de cada trabajo map/reduce, que decisiones se tomaron y con qué objetivos. Además de alguna alternativa de diseño que se evaluó y descarto, comentando el porqué.
    - El análisis de cada proceso corriendo en clusters variando la cantidad de nodos e indicando cuál sería la cantidad de nodos mejor para cada uno. Este análisis será realizado contra un archivo a determinar por la cátedra.

Entregas

- El día 09/11/2015 debe entregarse los nombres de los integrantes de cada equipo.
- El día 16/11/2015 a las 18 hs. Debe estar subido a la carpeta personal de uno de los integrantes del equipo el código fuente del trabajo. Se le indicará a la cátedra cual carpeta al ingresar a la clase es día.
- El día 16/11/2015 durante la clase cada grupo tendrá 15 minutos para configurar su entorno y mostrar la ejecución de la aplicación a la cátedra realizando dos corridas con de una de las opciones, con 2 archivos diferentes a determinar por la cátedra. Se tomará nota de las respuestas y los tiempos de ejecución y esto será parte de la evaluación del trabajo. A criterio de la cátedra también se podrán realizar preguntas sobre la implementación de los mismos
- El dia 16/11/2015 al finalizar la clase se deberá entregar el documento con el informe solicitado subiendolo a sakai.
- El día 30/11/2015 se entregarán las notas. Si se requiere que se corrija algún elemento del proyecto a modo de recuperar el trabajo deberá ser entregado al fin de la clase de este día.
- No se aceptarán entregas fuera de los días y horarios dados.

Sugerencia de investigación.

Pueden resultar útiles para la resolución del trabajo práctico investigar los siguientes temas:

- Hazelcast Collator.
- Jackson Custom Serializers (JsonDeserializer y JsonSerializer).