



Universidad Francisco Marroquín
Facultad de Ciencias Economicas
Computer Science
Algoritmos y Complejidad
Prof. Ernesto Rodríguez - erodriguez@unis.edu.gt
Aux. Juan Roberto Alvarado

Recuperación Parcial #1

Fecha de entrega: 04 de Octubre, 2018 - 11:59pm

Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a través de Github, en su repositorio del curso, colocado en una carpeta llamada "Recuperación 1". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex.

El objetivo de este trabajo es implementar el algoritmo de ordenamiento *merge sort* utilizando un framework de computación distribuida. Tiene la libertad de escoger el framework y lenguaje, a continuación se presentan algunas sugerencias:

- Cloud Haskell (Haskell)
- Orleans (.Net)
- Dispy (Python)
- Erlang (Erlang)
- MPI (c/c++)
- Akka (JVM)

Descripción

Su programa debe consumir una lista de oraciones en un documento de texto. Cada oración ocurrirá en una línea diferente. El objetivo es distribuir la ejecución de merge sort en varios procesos o computadoras debido a que los frameworks de computación distribuidas son transparentes ante el internet.

Al final de la ejecución de su programa, se debe producir una lista que tenga todas las oraciones de la lista original ordenadas alfabéticamente por el código ASCII. Puede asumir que solo se utilizarán letras (mayúsculas y minúsculas), números y espacios (no habrá signos de puntuación).

Evaluación

Una implementación funcional de este algoritmo con ejecución distribuida sera merecedora de un (30%) adicional en su nota del examen parcial. Adicionalmente, los siguientes criterios seran evaluados para puntos extra:

- Lectura en paralelo de la lista de oraciones
- Escritura en paralelo de la lista de oraciones
- Criterios arbitrarios de ordenamiento
- Utilización eficiente de nodos. En otras palabras, distribuir la carga optimamente según el numero de nodos que se disponga para el procesamiento.
- Interfaz web o grafica que facilite la utilización de su algoritmo
- Lectura distribuida de archivos. En otras palabras que los nodos puedan leer parte de la data localmente y retornar un resultado consolidado al nodo principal.