

Procesamiento paralelo

Crear un programa que permita calcular en concurrencia la media, la desviación estándar, el conteo, el valor mínimo y máximo de las series de datos de los archivos provistos

Este proyecto proporciona 1000 archivos con diferente data, pero la misma estructura que deben de ser procesados para calcular las funciones estadísticas solicitadas

Archivos de entrada:

	dates	Open	High	low	close
0	1970-01-01	92	371	92	240
1	1970-01-02	175	650	152	432
2	1970-01-03	933	933	933	933

Archivo de salida a generar (El mismo nombre archivo_**out**.csv):

	Open	High	low	close
count	16928.000000	16928.000000	16928.000000	16928.000000
mean	501.447011	667.695121	333.848594	501.771798
std	286.923979	233.547312	233.200717	227.144397
min	0.000000	1.000000	0.000000	1.000000
max	999.000000	999.000000	989.000000	997.000000

Condiciones:

- El programa podrá ser desarrollado en C++, Python, Java, c#
- Debe de ser ejecutado en base a la imagen de un contenedor (la imagen debe de poder descargarse de docker)
- Entregar el repositorio de GIT donde este implementado el código
- La imagen debe de ser ejecutada en docker con 1GB de RAM
- Los archivos deberán de ser procesados en paralelo como las funciones estadísticas
- El programa deberá de generar el archivo resultante de cada archivo existente en el folder
- Deberá de realizar múltiples corridas (10 iteraciones x modelo) restringiendo el uso de cores de la CPU
 - o 1 core – 1 hilo
 - o 1 core – 4 hilos

- 2 core 2 hilos
 - 2 core 4 hilos
 - 2 core 8 hilos
 - 4 core 8 hilos
- Deberá de generar un archivo con el track del tiempo que demoró el proceso en sus diferentes modelos
 - modelo de paralelismo
 - Todo secuencial
 - Paralelizando files (funciones estadísticas secuenciales)
 - Paralelizando funciones (archivos secuenciales)
 - Paralelizando Files y funciones (archivos y funciones en paralelo)

Entregable:

El estudiante deberá hacer un informe (XLS) con la recopilación de tiempo de cada uno de los problemas descritos y presentarlos en múltiples gráficas

- Grafica 1: contrastar el Tiempo total del proceso para los 6 escenarios del ejercicio secuencial
- Grafica 2: contrastar el Tiempo total del proceso para los 6 escenarios del escenario paralelizar los archivos
- Grafica 3: contrastar el Tiempo de los archivos para los 6 escenarios de paralelizar las funciones
- Grafica 4: contrastar el Tiempo de las funciones para los 6 escenarios de paralelizar archivos y funciones

Preguntas a contestar en clase: (llevar preparado los documentos para responder)

- ¿Cuál es el modelo de paralelismo más rápido en los 6 escenarios
- ¿Cuál opción modelo de paralelismo tomaría usted y por qué?
- ¿Recomendaría paralelizar tanto archivos y funciones al mismo tiempo?
- ¿Cuál es el factor de mejora de pasar de 1 Core a 2 Core para el proceso que paraleliza los archivos
- Determine el factor teórico de mejora para el escenario de 2 Core (amdahl's law) al paralelizar por archivo
- Determine el factor teórico de mejora para el escenario de 2 Core (amdahl's law) al paralelizar por función estadística
- Determine el factor teórico de mejora para el escenario de 2 Core (amdahl's law) al paralelizar por función estadística y archivos