Departamento de Ingeniería Informática
Sistemas Distribuidos
Manuel Ignacio Manríquez - Ariel Madariaga

Laboratorio 1 - MPI

Objetivos

1. Comprender y utilizar herramientas de comunicación para cómputo paralelo como MPI.
2. Diseñar e implementar una solución de cómputo paralelo distribuido para el procesamiento de datos.
3. Implementar y comparar múltiples estilos arquitecturales en el contexto de un problema de cómputo paralelo.

The One Billion Row Challenge

A finales del año 2023, *The One Billion Row Challenge* (1brc, de ahora en adelante) fue propuesto a la comunidad de usuarios de Github para determinar qué tan rápido puede ser el código Java moderno.

Aunque el desafío fue propuesto inicialmente solamente para los usuarios de Java, rápidamente se empezaron a publicar resultados en otros lenguajes populares: Rust, C++, Go y otros.

Los requisitos funcionales son sencillos. Se tiene un archivo de entrada donde cada fila corresponde a una medición de temperatura proveniente de alguna estación meteorológica determinada. Se deben recorrer todos los registros para obtener la temperatura mínima, máxima y promedio para cada estación. Cada línea del archivo de entrada contiene el nombre de la estación y la temperatura medida, separados por un punto y coma: `<string estación>;<double temperatura>.`

Una porción de archivo de entrada se vería de la siguiente manera:

```
Hamburg;12.0
Bulawayo;8.9
Palembang;38.8
```

```
St. John's;15.2
Cracow;12.6
Bridgetown;26.9
Istanbul;6.2
Roseau;34.4
Conakry;31.2
Istanbul;23.0
```

Los archivos de entrada serán progresivamente más grandes - hasta superar los mil millones de registros por archivo.

Además de ser un interesante ejercicio de laboratorio, el procesamiento de grandes volúmenes de datos es una aplicación de sistemas distribuidos con múltiples aplicaciones en múltiples industrias: procesamiento y análisis de transacciones financieras, predicción del clima, simulaciones de fenómenos físicos de alto costo computacional, etc.

A pesar de que la mayoría de los resultados propuestos, incluso los publicados por ingenieros en otros lenguajes de programación, a nosotros nos interesa aprender cómo se comportan las soluciones basadas en principios de diseño de sistemas distribuidos. En este laboratorio implementaremos distintos enfoques para la resolución de estas agregaciones en conjunto masivos de datos, compararemos su comportamiento y derivaremos conclusiones.

Indicaciones

Para el desarrollo del laboratorio, tengan en cuenta las siguientes consideraciones:

1. El laboratorio debe realizarse en grupos de una o dos personas - no más.
2. La calificación final de este laboratorio depende de un repositorio y de un informe, que deben ser entregados al profesor y ayudante dentro de los plazos establecidos.
3. Los repositorios deben ser públicos una vez terminado el plazo de entrega, o deben invitar al profesor y ayudante el repositorio privado. De lo contrario, este componente de la evaluación será considerado con la nota mínima.
4. Se realizarán descuentos por malas prácticas de programación y por mala redacción u ortografía.

5. El repositorio debe incluir un archivo README que contenga las instrucciones de instalación, en conjunto con las versiones de los componentes de software utilizados.
6. Los diseños de arquitectura deben considerar tanto el diseño de software como el diseño de sistemas. Estos deben estar correctamente justificados (por qué creen que es el mejor enfoque al problema y las fuentes que lo respaldan).
7. El plazo de entrega termina a las 23:59 hrs de la fecha de entrega especificada.
8. El procesamiento y la comunicación entre nodos procesadores debe realizarse en Python y usando MPI4py.

Para el desarrollo del laboratorio y la obtención de resultados, deben proponer e implementar tres soluciones con distintos enfoques arquitecturales:

1. Monolítico (no distribuido): Esta solución corresponderá a la línea base sobre la cual se compararán los resultados obtenidos posteriormente. Aunque puede ser multihebreada, se debe restringir a un solo proceso físico.
2. Basado en servicios: Una solución donde la solución está distribuida en múltiples servicios, los cuales interactúan directamente entre sí. Cada servicio debe vivir residir en su propio proceso, aunque son libres de implementar múltiples hebras.
3. Basado en eventos: Los servicios o procesos que componen la aplicación emiten eventos, los cuales son procesados y enrutados por un middleware.

Cada solución debe considerar una capa de presentación (puede ser una simple interfaz gráfica) y una capa de persistencia (para almacenar los resultados y acceder a los datos de entrada). El cómo implementar estas capas y en qué tecnologías hacerlo queda a discreción de los estudiantes.

En adición a lo anterior, el informe a presentar debe considerar al menos los siguientes puntos:

1. Introducción: una breve descripción del documento e introducción al contexto del problema.
2. Diseños de solución: en esta sección, se deben presentar los diseños de solución previamente detallados. Es apropiado incluir diagramas, y fuentes que respalden sus decisiones de diseño. Para esto les podría resultar útil incluir casos de estudio relevantes.

3. Metodología: esta sección del informe debe detallar la metodología utilizada para obtener los resultados presentados. Recuerden que sus resultados deben ser reproducibles, mucho de lo cual depende del nivel de detalle presentado en esta sección. Se deben considerar configuraciones de hardware y archivos de entrada, cantidad de ejecuciones para la obtención de métricas, etc.
4. Resultados y discusión: se presentan y discuten brevemente los resultados obtenidos. Es apropiado describir las posibles motivos de los resultados, como costos en los que incurre una solución y no otra, etc.
5. Conclusiones: se presentan conclusiones en base a los resultados obtenidos. Aquí es apropiado considerar otros factores de análisis relevantes para una implementación real de una solución de este tipo: escalabilidad (de todo tipo), costos organizacionales y ventajas y desventajas.

La calificación se obtendrá mediante un promedio simple entre la nota del código y el informe. Ambas calificaciones deben ser mayor o igual a 4.0, de lo contrario, la calificación final será el mínimo entre ambas.

Fecha de entrega: domingo 28 de abril.