Trabajo 2 – Computación en Paralelo

Indicaciones Generales:

1. **Formato de trabajo**:
   - Este trabajo se realizará en grupos armados por los propios estudiantes.
   - El trabajo se entregará en **un (1) Jupyter Notebook (Parte I, III y IV)** y **dos (2) archivos de Python (Parte II)** con los códigos utilizados para resolver las preguntas. Los códigos deben ser claros, ordenados y suficientemente comentados para que otra persona pueda reproducirlos fácilmente.
   - Los comentarios deben indicar claramente cada subsección, así como explicar los pasos seguidos.
   - Solo un miembro del equipo deberá enviar por correo (vfuentes@pucp.edu.pe) el "Contrato de grupo de trabajo" (disponible en Canvas) y el archivo de solución del trabajo.
2. **Evaluación**:
   - Se valorará la claridad, replicabilidad y orden del código.
   - El informe debe ser claro y bien estructurado, con respuestas concisas.
3. **Materiales de referencia**:
   - Se deberán utilizar los comandos revisados en clase.
4. **Fecha de entrega**:
   - Límite: Lunes 27/01 a las 11:59 p.m. La demora en el envío se penalizarán con cinco puntos por cada cuatro horas de tardanza.

Parte I (5 puntos)

Responda las siguientes preguntas en máximo 50 palabras.

1. ¿Qué es un proceso embarrassingly parallel y uno inherentemente serial? Dé un ejemplo de cada uno (distintos a los vistos en clase)
2. ¿Cuáles son los dos principales cuellos de botella al paralelizar un proceso? Explíquelo a partir de las leyes de Amdahl y Gustafson
3. Describa los recursos (CPU y GPU) de su computadora y provea evidencia (ie. screenshot de la computadora de un integrante del equipo).
4. ¿Cuál es la diferencia entre point-to-point communication y collective communication en MPI? ¿Cuál es la diferencia entre las operaciones de broadcasting, gathering y scattering?

Parte II (5 puntos)

Escribir un código ("parte2_1.py") que realice:

a. Que un procesador genere una lista y lo envíe a otros tres procesadores.
b. Que cada uno de los tres procesadores reciba la lista enviada, imprima su número de procesador y la lista.
c. En otro chunk responda: De ejecutar el código 100 veces, ¿el orden de los resultados será siempre igual? ¿Por qué?

Escribir un código ("parte2_2.py") que realice lo siguiente:

d. Usando la opción de broadcasting en MPI, defina un diccionario de cuatro elementos que contenga sus cuatro cursos favoritos del diplomado desde el primer procesador. Repita esto para todos los procesadores disponibles. Luego, registre y compare el tiempo de demora desde cada procesador.
e. Defina una secuencia de valores: {0,1,2,3,…, n} en donde n es el número de procesadores de su computadora desde uno de los núcleos. Luego disperse los valores a cada uno de los núcleos restantes usando la opción scattering. Identifique si el número asignado por el proceso coincide con el rango del procesador

---

Parte III (10 puntos)

***Atención**: Su aplicación debe tener por lo menos dos variaciones de fondo respecto a los notebooks de ejemplo vistos en clase y del portal oficial de Dask. Podrá utilizar datos de cualquier fuente, pero para algún problema o pregunta ocurrido en el Perú.*

Para esta parte los grupos deberán realizar una aplicación de Machine Learning (ML). El tema del ejercicio es libre, pero debe ser propio y de interés del grupo (la justificación del tema será evaluada).

Deberá presentar este ejercicio en un Jupyter o Colab Notebook (.ipynb) donde todas las celdas ya hayan sido ejecutadas; además del mismo documento en formato PDF. Este deberá incluir las siguientes partes:

1. Presentación y relevancia del problema de predicción que se desea realizar (por ejemplo, predecir la vulnerabilidad de los hogares en el Perú). Defina claramente qué datos utilizara, cuál es su variable target y cuáles sus predictores.
2. Describa los pasos a realizar para su aplicación de ML.
3. Describa cómo llevaría a cabo este ejercicio de manera paralelizada y compárelo a su aplicación serial. Como parte de esta descripción, incluir los siguientes aspectos:
   a. Explicar qué partes del ejercicio se harán de forma serial y por qué no paralelizo estas tareas.
   b. Para las tareas en paralelo, explique usando el método de Foster como se dan las etapas de partición, comunicación, aglomeración y mapeo (PCAM) para su aplicación.
   c. Discuta qué tipos de procesadores podría utilizar para cada parte. (No es necesario que utilice los GPU pese a que señale su mejor desempeño)

   d. Identificar los cuellos de botella del ejercicio y comente hasta qué punto la paralelización puede ayudar a resolverlos.
4. Uso de Dask.
   a. Ejercicio de ETL: Usar Dask Dataframes para cargar la(s) base(s) de datos que se utilizará(n) y presentar lo siguiente:
      - Creación de por lo menos dos variables
      - Por lo menos dos estadísticos descriptivos que vayan en línea con el tema y argumento. Explíquelos
      - Por lo menos dos gráficos que vayan en línea con el tema y argumento.
      - Tanto los gráficos como los descriptivos deben estar en calidad para ser incluidos en un reporte final. Se descontarán puntos por presentación descuidada.
   b. Implementación de Machine Learning: Utilice Dask para entrenar por lo menos un modelo de Machine Learning supervisado. Este acápite debe contener, por lo menos, los siguientes elementos:
      - Definición de predictores (X) y vector de target (y)
      - Train-test splitting
      - Cross-Validation
      - GridSearch
      - Model fit y selección del modelo óptimo
      - Computo de dos indicadores de la calidad de ajuste en muestra entrenamiento
      - Computo de dos indicadores de la calidad de ajuste fuera de la muestra (test)
      - Limitaciones y posibles extensiones

Recomendaciones
- No se evaluará la complejidad de modelo de ML, al no ser la intención del curso. En realidad, lo importante es demostrar el dominio del uso de Dask para seguir todos los pasos de un pipeline de ML con paralelización.
- Se recomienda el uso de *dask_ml*. Sin embargo, también puede utilizar otras alternativas como, por ejemplo, *ray*.
- Si deciden utilizar datos provenientes de encuestas complejas, no es necesario que considere el diseño ni los pesos muestras en ninguno de los pasos a fin de no complejizar el ejercicio.