ID	PR - 20.05
Fase	Implementación
Cumplimiento	30%

Objetivos de la semana

- Avanzar en la implementación del solver manager.
- Integrar correctamente la nueva implementación con Pyomo.

Tareas realizadas

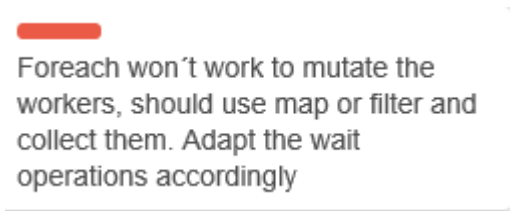
Corregir la instanciación del solver

En la inicialización de cada *PHSparkWorker* se instancia el solver a utilizar. Existía un error de dependencias que no eran transmitidas al runtime de Spark. Se solucionó haciendo un import de *pyomo.environ* previo a la instanciación del solver.

Commits

- Started implementing PHSparkWorker

Corregir implementación del RDD

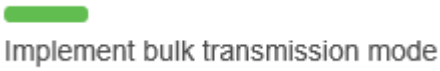


Inicialmente se intentó utilizar el método *foreach* para procesar información en los workers del RDD. Sin embargo esto no funciona porque el RDD se debe mutar de forma que, una vez realizada la iteración, se mantenga un RDD con los workers nuevos. Por esto, la implementación pasa a utilizar el método *map* donde, para cada worker, se procesará la iteración y se devolverá el worker con los datos actualizados.

Commits

- Worker queue

Implementar transmisión en bloque



Para poner tareas en la cola de trabajo de los workers, Pyomo puede hacerlo secuencialmente o agrupar varias tareas para ejecutarlas transaccionalmente.

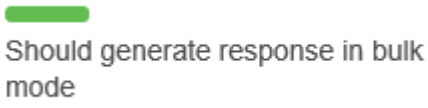
```
if self._bulk_transmit_mode:
    if queue_name not in self._bulk_task_dict:
        self._bulk_task_dict[queue_name] = []
    self._bulk_task_dict[queue_name].append(task)

else:
    self._rddWorkerList = self._rddWorkerList.map(lambda worker:_do_parallel_work(worker, task,
queue_name)).cache()
```

De esta forma, se acumulan varias tareas que se ejecutarán cuando se indique el final de la transacción llamando al método *end_bulk()*.

Commits

- Phspark result return

Guardar los resultados en el modo de ejecución transaccional

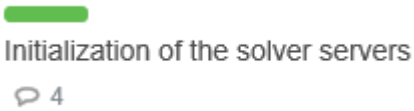


Los resultados del procesamiento en los workers no se devuelven directamente al invocador. Tras cada proceso, el worker guarda los resultados en una lista. En cualquier momento posterior se podrán solicitar estos resultados haciendo uso de los *action_handle* asignados a cada tarea.

Commits

- Phspark result return

Inicialización de los solver servers



Una vez implementada la integración de Spark con Pyomo, comprobar que los workers del RDD se inicializan correctamente.

Commits

- Spark workers initialized correctly

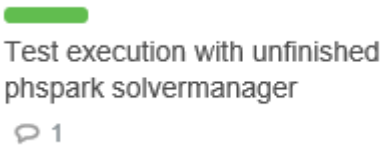
Corregir manejo de los action handle

El solver manager deberá guardar correctamente los *action handle* y las tareas que referencia cada uno. De esta forma, cuando se solicite un resultado se podrá localizar el worker que lo contiene y devolver el *action handle* con el resutlado correcto.

Commits

- Phspark result return

Probar lo implementado hasta ahora



Debemos comprobar que lo implementado hasta este punto funciona correctamente. Los workers deben estar inicializados correctamente y el RDD mantener la coherencia entre iteraciones.

Hay algunos problema con la serialización. Uno se soluciona eliminando el parámetro *_scenario_instance_factory* del solver server, pues sólo se utiliza en la inicialización y no será usado más tarde. Adicionalmente, el uso de funciones dentro de lambdas para el método *map* del RDD no funciona adecuadamente con el serializador por defecto. Como solución a esto último se utiliza *Cloudpickle* como serializador para Spark.

Se detecta que sucesivas iteraciones del algoritmo no funcionan adecuadamente, será necesario estudiar la persistencia de los datos en Spark.

Commits

- Changed default serialization library
- Testing spark execution