

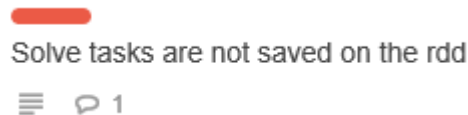
ID	PR - 08.07
Fase	Implementación
Cumplimiento	90%

Objetivos de la semana

- Solucionar los problemas de persistencia de los datos en Spark.
- Limpieza del código que funciona

Tareas realizadas

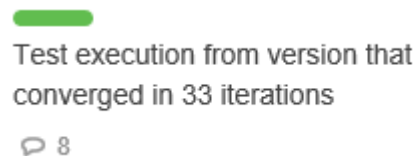
Las tareas "solve" no se guardan en el RDD



Se implementó un historial en los workers donde se van guardando las tareas realizadas. De esta forma, en el método "getstate" podemos ver si el RDD serializado es la versión correcta.

En una de las versiones probadas, la tarea "solve" no estaba siendo guardada, indicando algún tipo de error en la serialización causando que la siguiente iteración se ejecutase a partir de un RDD no actualizado.

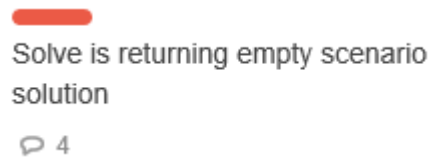
Tests con la versión que converge en 33 iteraciones



Se hacen pruebas con la versión anterior que converge en 33 iteraciones para hacer comparaciones e intentar localizar algún problema.

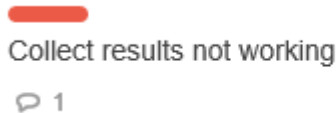
Comprobado que utilizando "local" converge y usando "local[4]" falla en la llamada a TransformationFactory. Basándose en esto se buscan objetos compartidos para descartar problemas de concurrencia pero no se llega a ninguna conclusión.

Método "solve" devuelve solución vacía

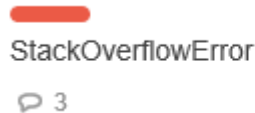


Tras el collect, los valores de solución para el escenario son vacíos. Comprobamos que al finalizar el "process" los valores son correctos, pero al deserializar ya están vacíos. Podemos entonces localizar el problema en la serialización.

Tarea adjunta

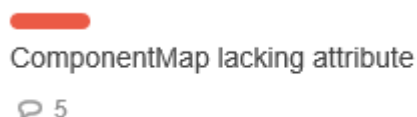


StackOverflowError



Cuando uno de los workers falla inesperadamente puede causar que en el collect se devuelva una lista vacía. En este caso el master seguirá solicitando resultados hasta recibir los correctos, causando que el RDD se recompute en un bucle infinito hasta que se llena la pila.

Atributo no encontrado en ComponentMap



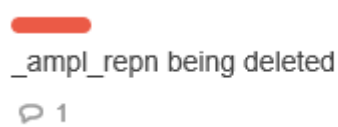
En "_preprocess_scenario_instances" se genera una lista de atributos para escribir el archivo de entrada para el solver. En la segunda iteración de "solve" esta lista no se crea correctamente y provoca una excepción.

Haciendo un seguimiento de su creación algunos de los objetos no se imprimen y se supone un problema de dependencias. Sin embargo es más probable que sea un problema al imprimir, pues otros atributos del objeto si se imprimen, afirmando que el objeto existe.

Commits

- [Debugging ComponentMap creation](#)

Atributo `_ampl_repn` borrado



Tras localizar que este atributo es el que se pierde entre iteraciones se hace un seguimiento del mismo. Encontramos que en `block.py::__getstate__` se está borrando deliberadamente en la serialización del objeto.

Comentando estas líneas el programa se ejecuta satisfactoriamente en spark.

```
# Note sure why we are deleting these...
if '_canonical_repn' in ans:
    del ans['_canonical_repn']
# if '_ampl_repn' in ans:
#     del ans['_ampl_repn']
return ans
```

Commits

- [Fixed execution](#)

Reuniones

El día 05/07 se realizó otra reunión para localizar el problema que impedía la ejecución del programa. Esta reunión fue finalizada solucionado el problema indicado previamente.