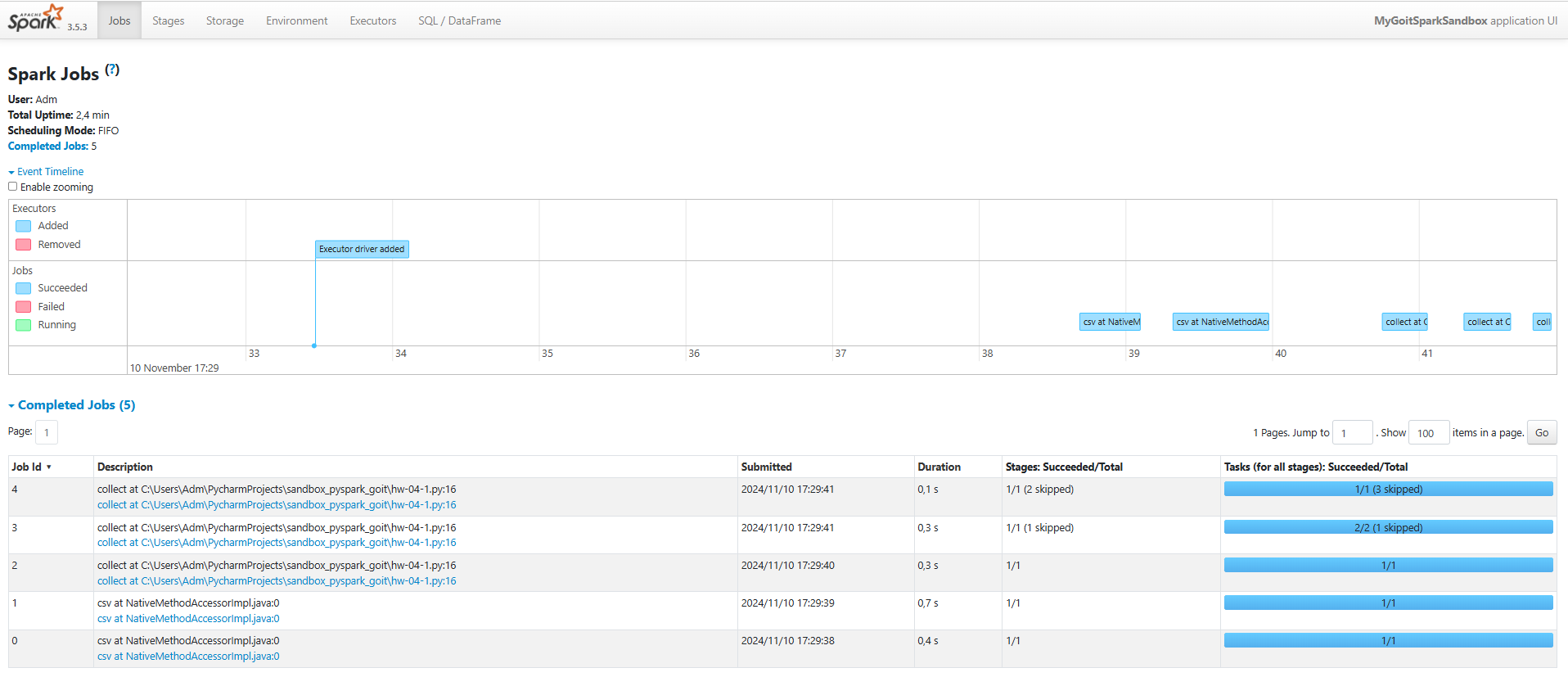
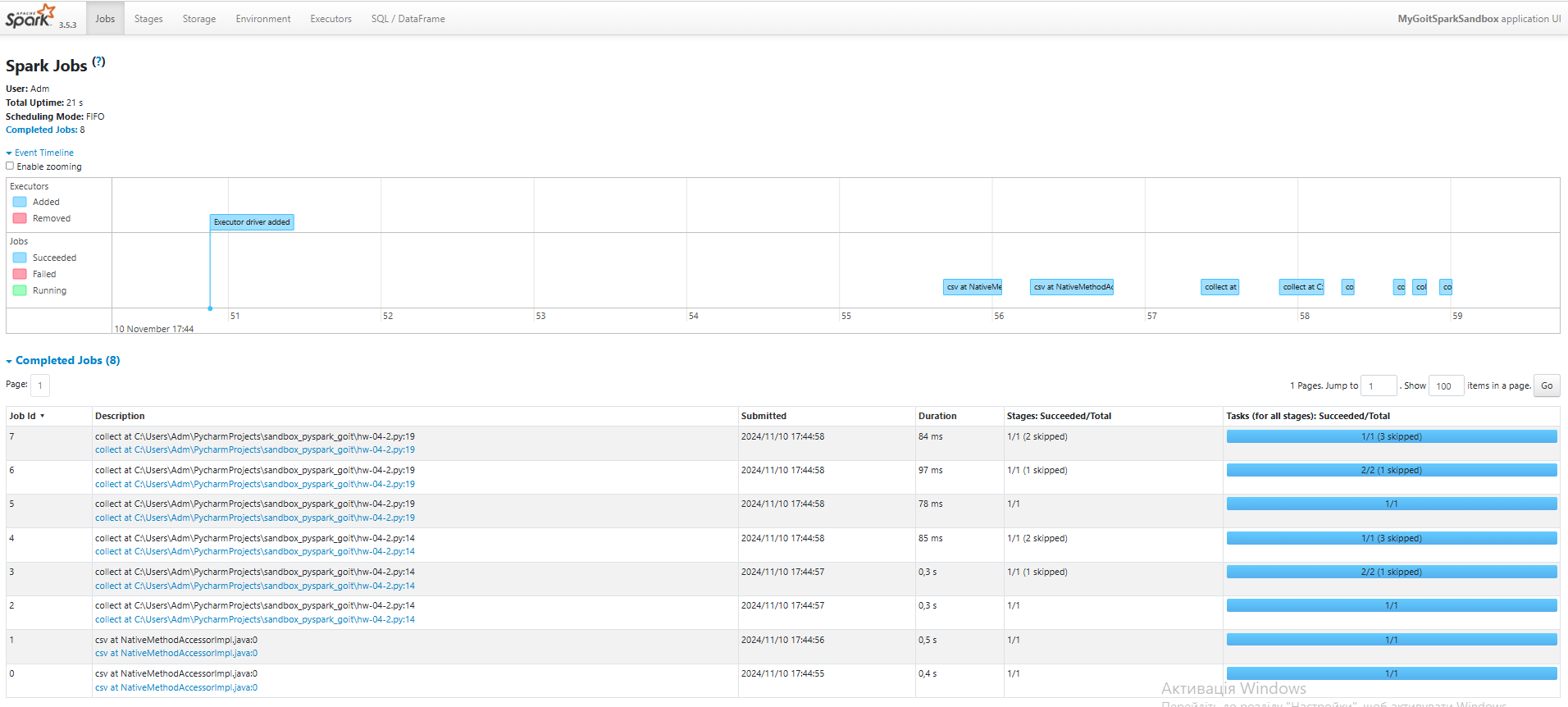
## **1.Результат виконання запиту з одним action'ом**



### **Jobs:**

0: Сканування даних;  
1: Job 0 + Десеріалізація;  
2: Репартиціонування;  
3: Фільтрація.Групування.Підрахунок рядків;  
4: Збирання результату.

## **2.Результат виконання запиту з додатковим action'ом**



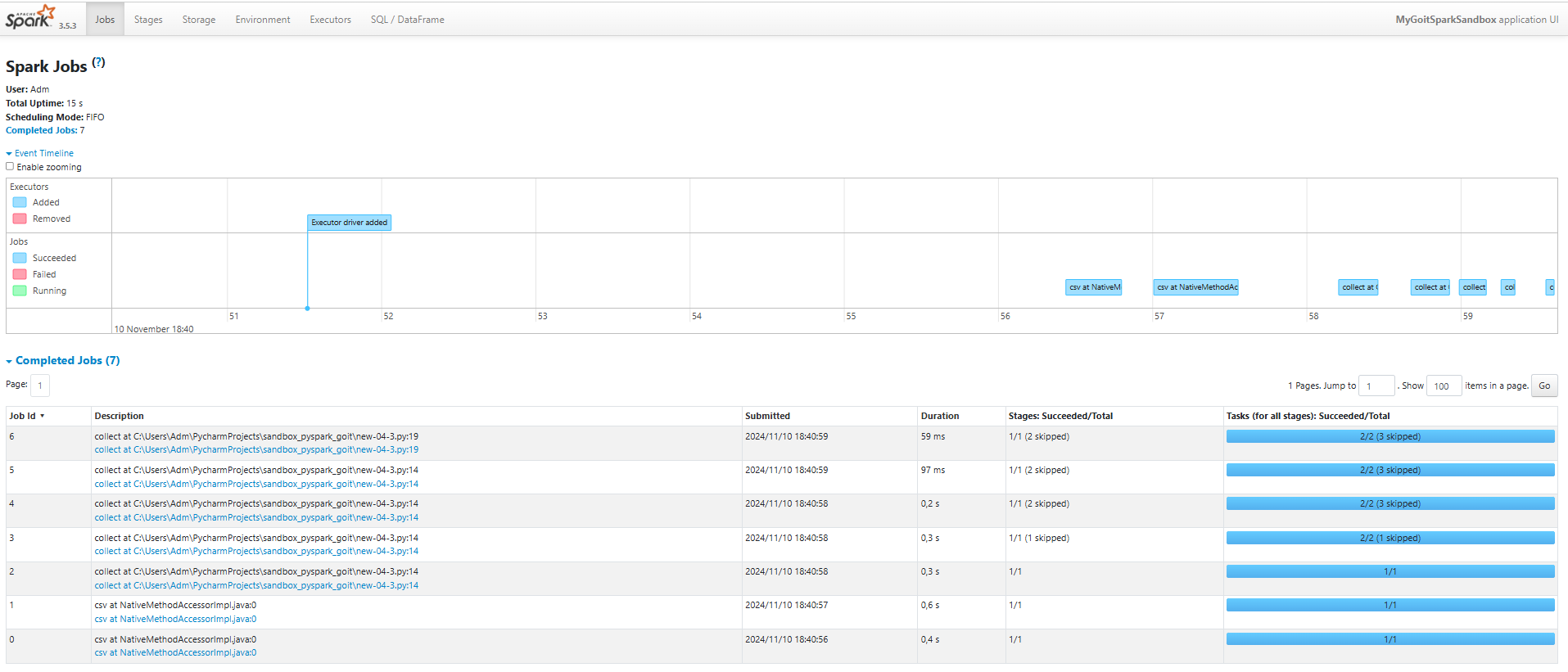
### **Jobs:**

0: Сканування даних;  
1: Job 0 + Десеріалізація;  
2: Репартиціонування;  
3: Фільтрація.Групування.Підрахунок рядків;  
4: Проміжне збирання результату;  
5: Репартиціонування;  
6: Фільтрація;  
7: Збирання результату.

#### **Висновок:**

Після проміжного collect(), коли ми додали нову трансформацію where("count > 2"), Spark виконує попередні трансформації знову, якщо результат не був кешований.

## **3.Результат використання функції cached()**



### **Jobs:**

0: Сканування даних;  
1: Job 0 + Десеріалізація;  
2: Репартиціонування;  
3: Фільтрація.Групування.Підрахунок рядків.Кешування даних;  
4: Проміжне збирання результату;  
5: Фільтрація;  
6: Збирання результату.

#### **Висновок:**

Використання cache() у Spark дозволяє зберігати результат проміжних трансформацій у пам'яті, що дозволяє уникнути повторного виконання цих трансформацій при наступних діях. Це зменшує кількість джобів, оскільки Spark не потрібно повторно виконувати ті ж самі трансформації