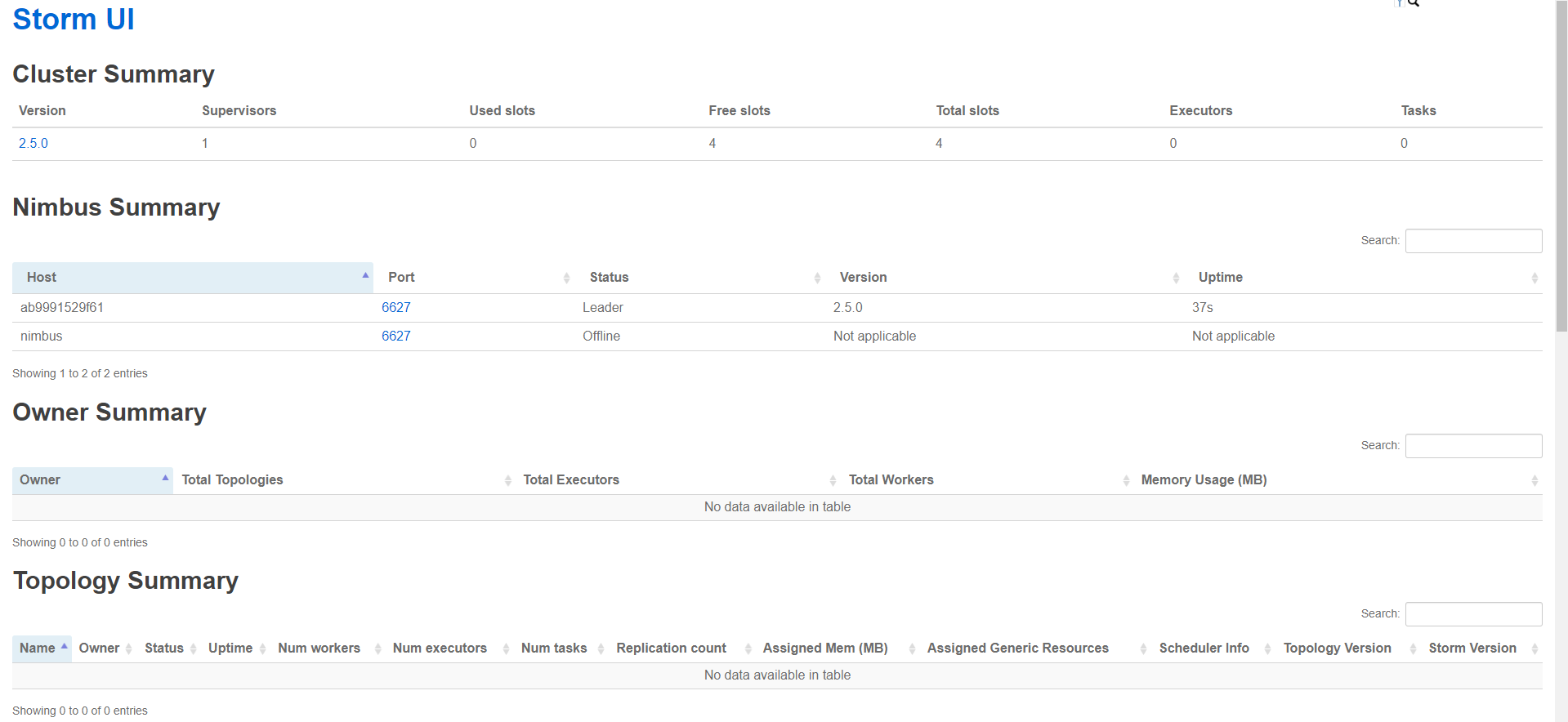
Docker

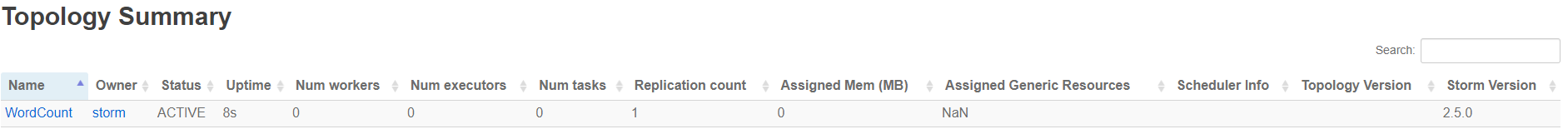
2.1)



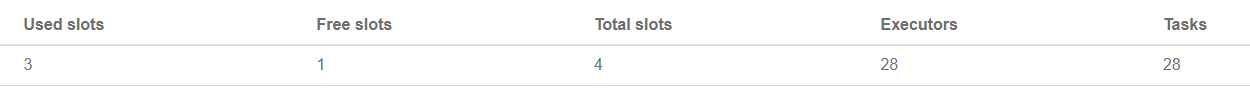
2.2) Car on a pas encore créer de topologie (cf Antoine)

WordCount

2.3)

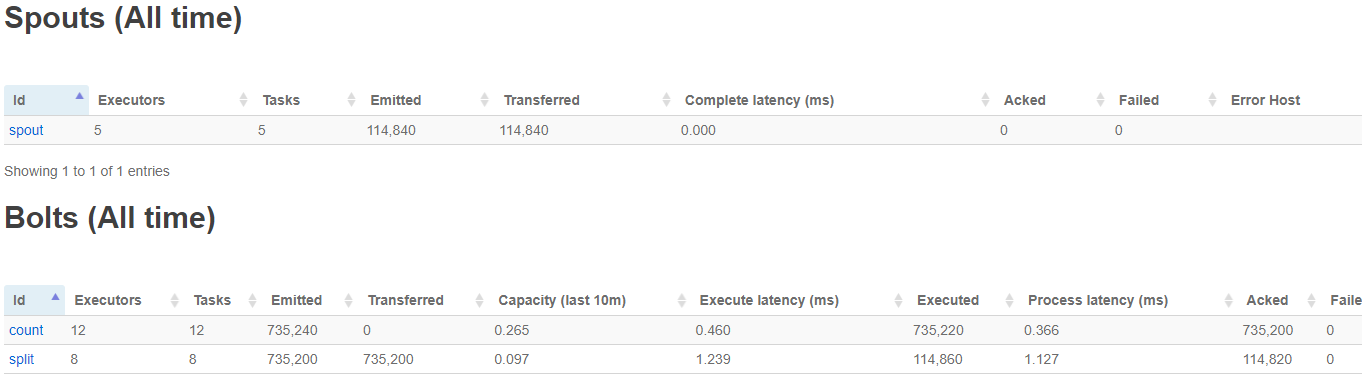


2.4)

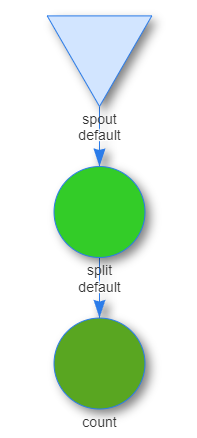


3 slots utilizes et 28 executors

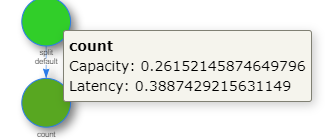
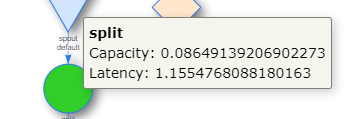
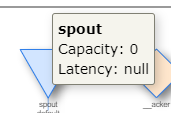
1.1)

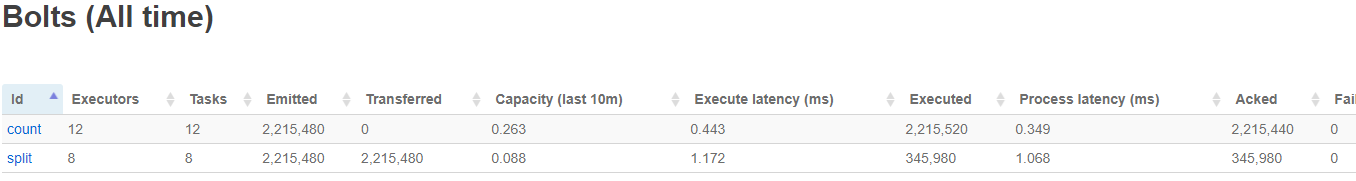


2)

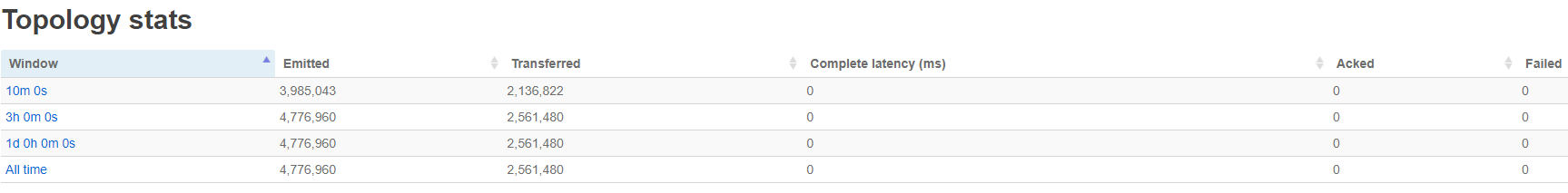


2.1)





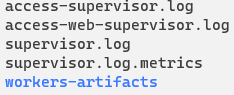
2.2)



Emitted : number of tuples emitted

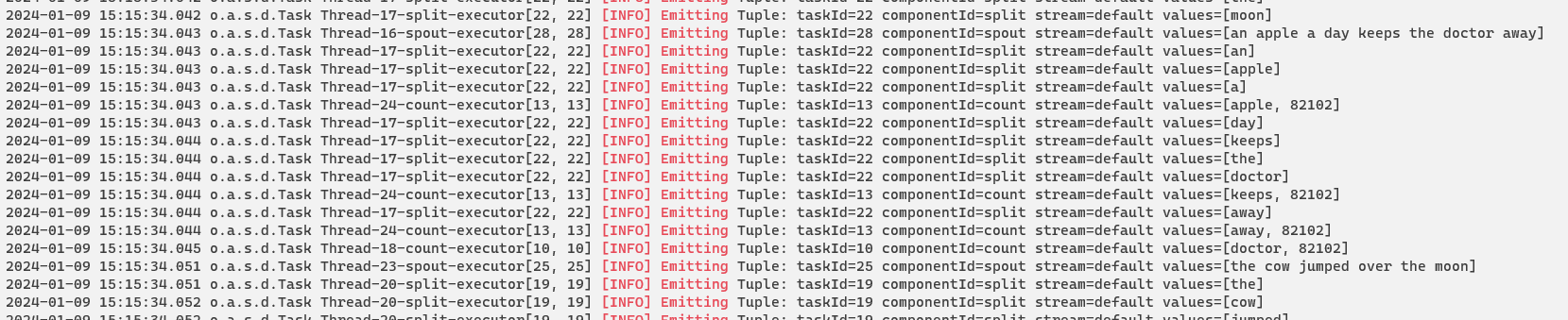
Transferred : number of tuples emitted that send to one or more bolt

2)



3)

4)

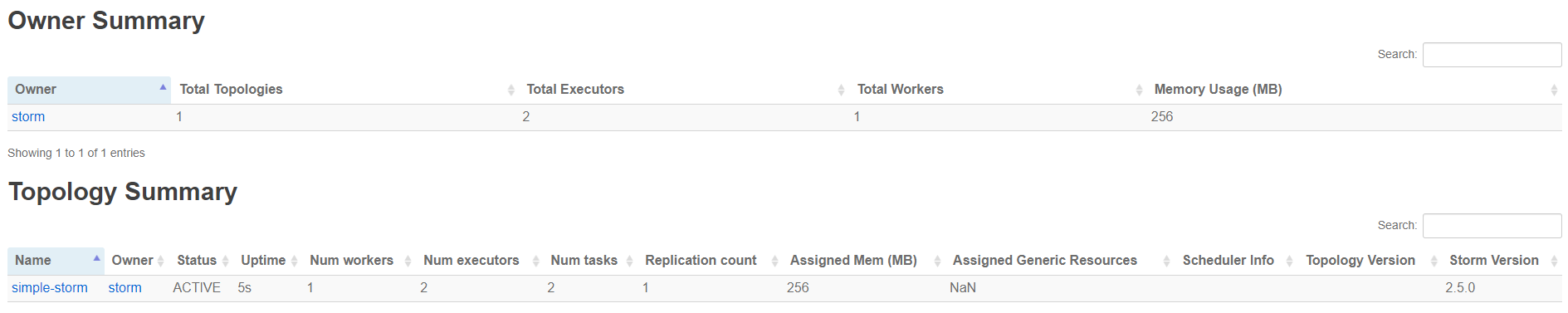


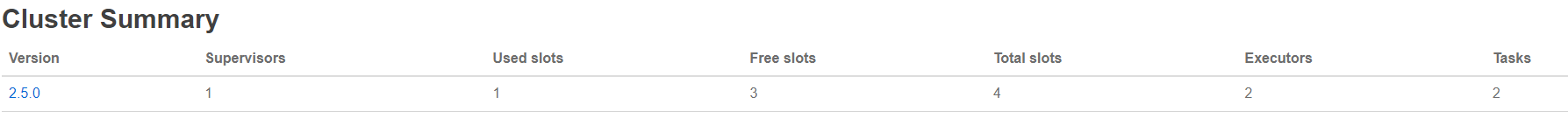


Simple storm

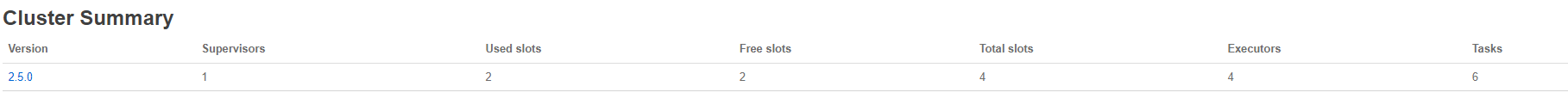
**Random spout**

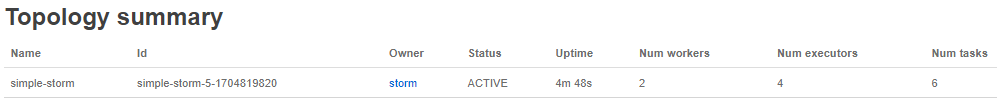
storm jar /storm/labs/storm-simple-2.6.0.jar org.example.Main SimpleStorm





1 worker (JVM), 2 executor (1 spout et 1 \_acker)

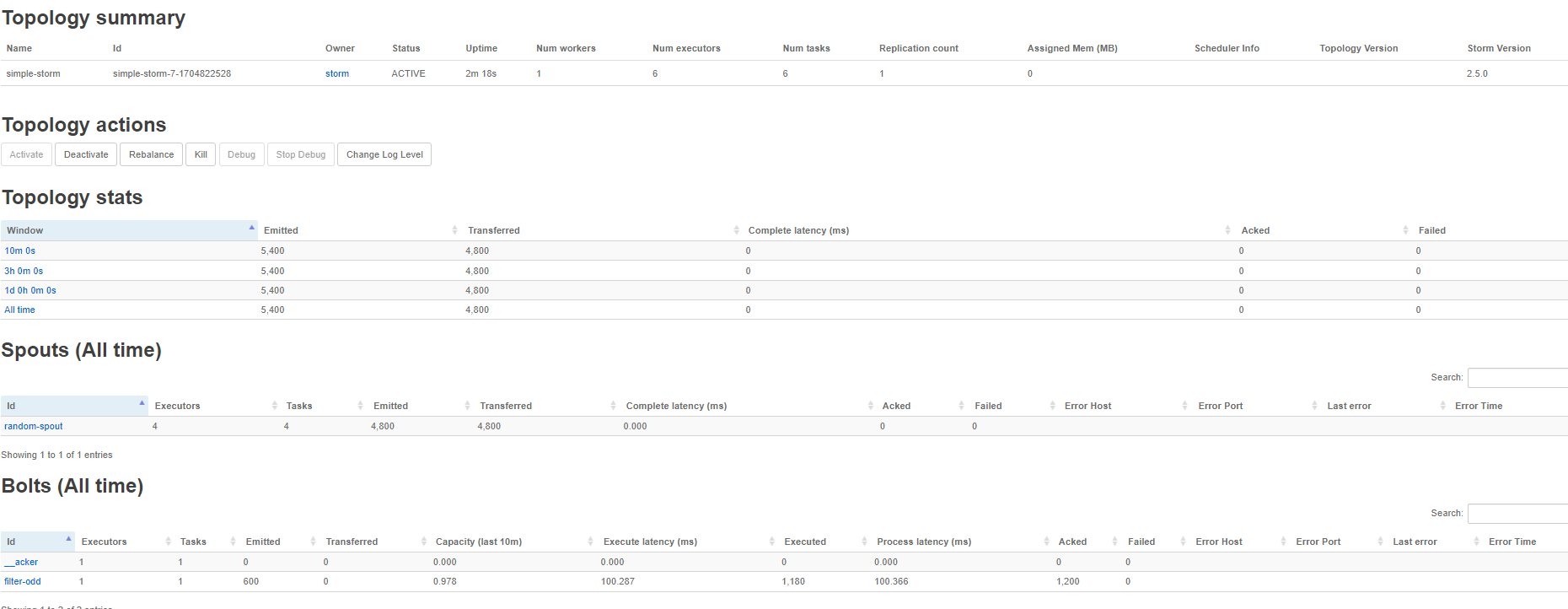




2 worker (JVM), 4 executors (4 spout) et 6 task (4 spout et 2 acker)

**Filtering bolt**

Bolt et spout ne sont pas sur le meme executor



Filter -> sleep 100ms

Spout -> sleep 100ms

600 \* 2 (add even number) \* 4 (num spout) = 4800

Le filter est surchargé et ne peut donc emettre que 1/8 eme des données recus. Il traite effectivement 4 fois moins vite que les spout. Les tuples ne seront jamais tous traité et ceux traité le seront plus lentement. Il y aura une grande latence entre les elements créés et quand ils seront traités.

**Averaging bolt**

**Log Analysis**



Date (mois, jour, heure) | Nom de la machine | numéro de la connections sh ???

DAG :

* LogSpout : get text from disk file

storm jar /storm/labs/storm-simple-2.6.0.jar org.example.log.Main LogStorm