**Synthèse: Ressources Machines pour l'entraînement du Chatbot**

Ce document contient un bilan de l’étude sur l’utilisation des ressources machines pour l’entraînement d’un chatbot avec RASA NLU. Les tests qui ont permis de relever toutes les valeurs ont été réalisés sur trois machines différentes, avec le même jeu de données et en utilisant les mêmes configurations RASA pour chaque test.

*Des vidéos de l’évolution de l’utilisation des ressources machines durant l’entraînement ont été réalisé, et seront fournis avec les livrables.*

**==================================================================**

Le jeu de données qui a servi de base d’entraînement pour les trois entraînements, effectués sur trois machines différentes, est un fichier json contenant des exemples d’annotations selon le format RASA NLU. **La taille du fichier est 6 Mo** (~1500 exemples).

Pour les trois entraînements, le pipeline utilisé est le suivant:

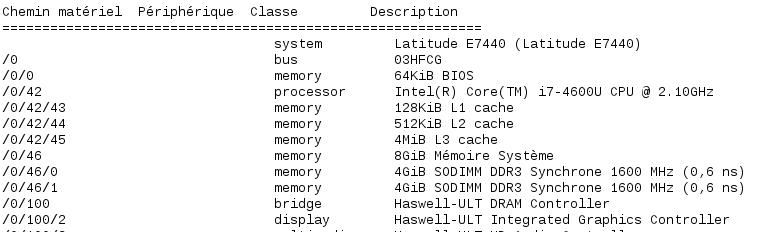
>>

***language: "fr"  
pipeline:  
- name: "nlp\_spacy"  
- name: "tokenizer\_spacy"  
- name: "intent\_entity\_featurizer\_regex"  
- name: "intent\_featurizer\_spacy"  
- name: "ner\_crf"  
- name: "ner\_synonyms"  
- name: "ner\_spacy"  
- name: "intent\_classifier\_sklearn"  
- name: "ner\_duckling\_http"  
 # url of the running duckling server  
 url: "http://0.0.0.0:8000"  
 # dimensions to extract  
 dimensions: ["time", "Quantity", "Numeral"]  
 # allows you to configure the locale, by default the language is  
 # used  
 locale: "fr\_Nothing"  
 # if not set the default timezone of Duckling is going to be used  
 # needed to calculate dates from relative expressions like "tomorrow"  
 timezone: "Europe/Berlin"***

>>

**Machine 1**

* **Caractéristiques**

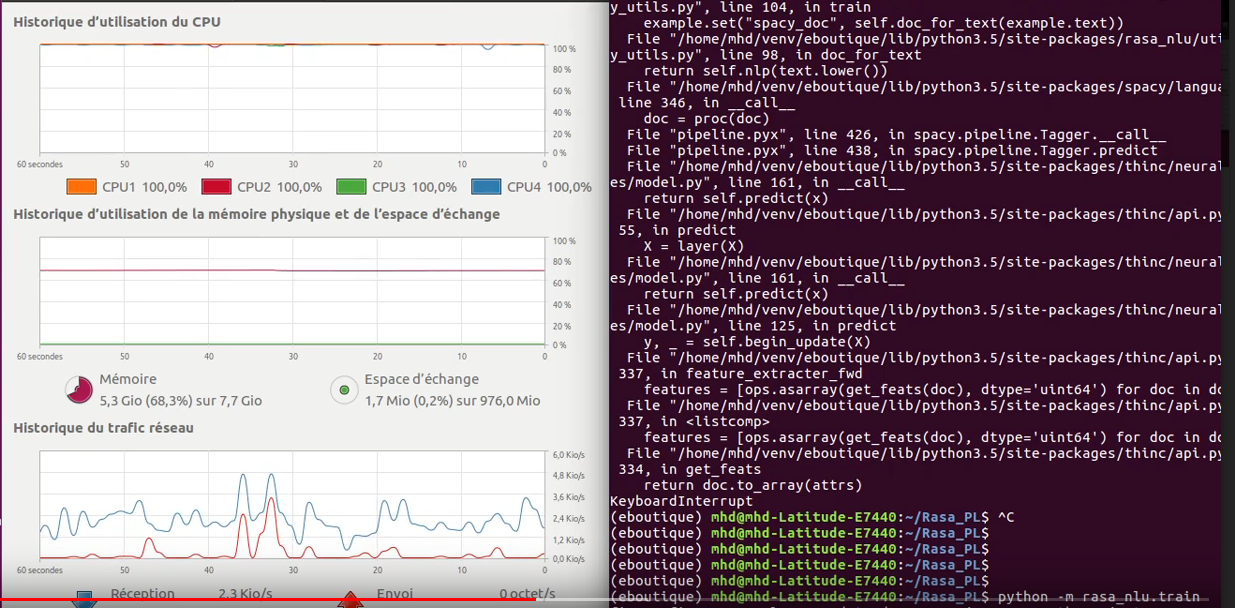


* **Performances**

**Durée entraînement:** 10 min

**Utilisation RAM max:** 1.8 Go

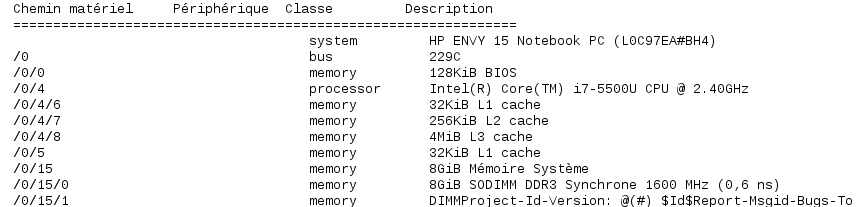
**Utilisation CPU max:** 4 CPU à 100 %



**Vidéo: *m1\_perf.mkv***

**Machine 2**

* **Caractéristiques**

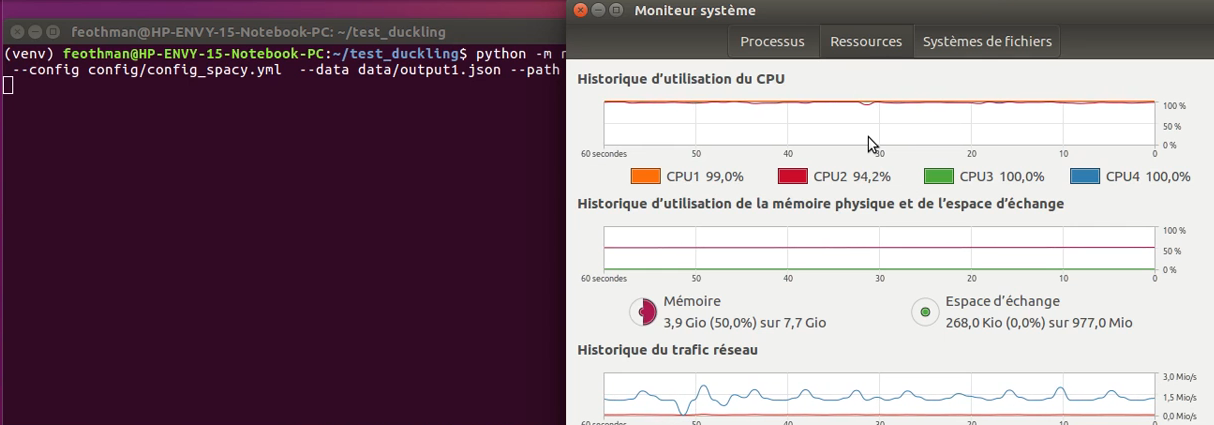


* **Performances**

**Durée entraînement:** 13 min

**Utilisation RAM max:** 2.5 Go

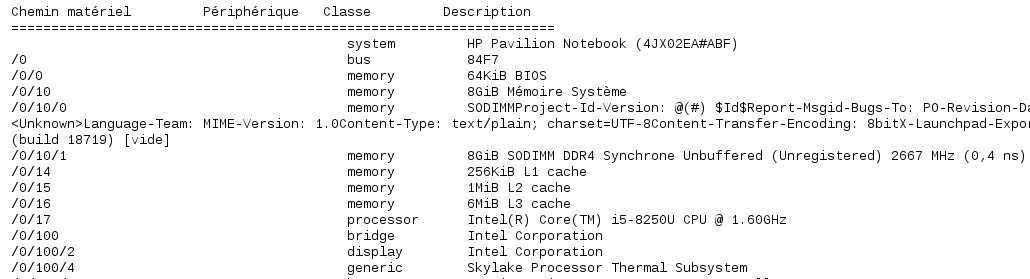
**Utilisation CPU max:** 4/4 CPU à 100 %



**Vidéo: *m2\_perf.mkv***

**Machine 3**

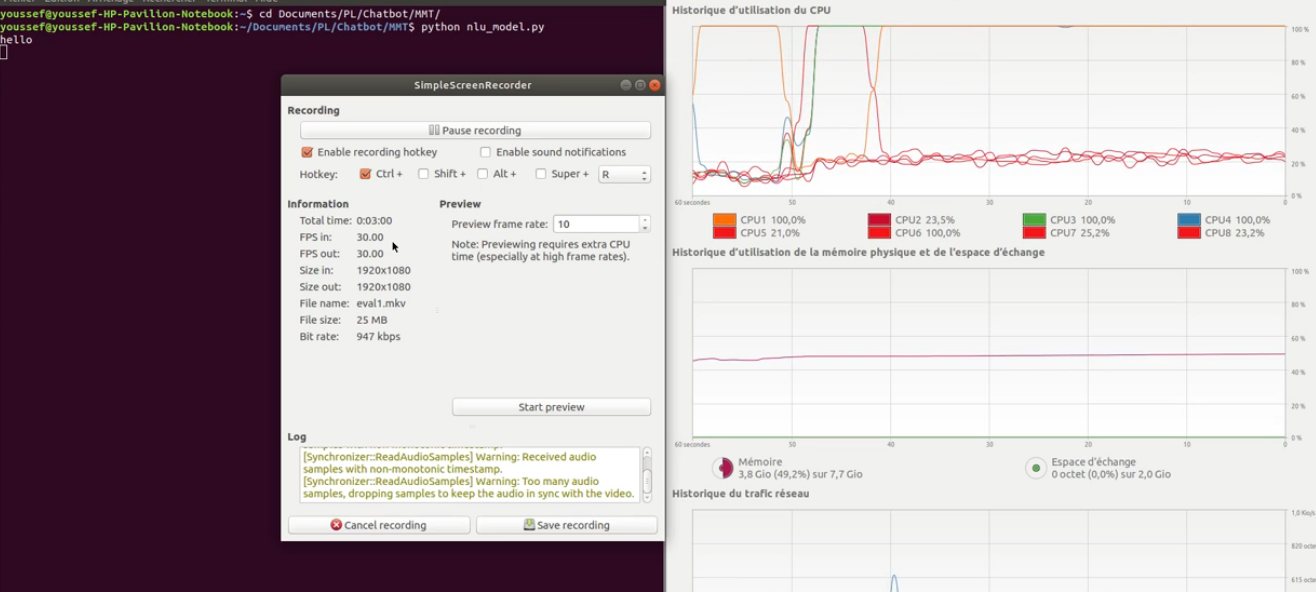
* **Caractéristiques**



* **Performances**

**Durée entraînement:** 6 min

**Utilisation RAM max:** 2.4 Go

**Utilisation CPU max:** 4/8 CPU à 100 %

**Vidéo: *m3\_perf.mkv***

**==================================================================**

**Bilan**

L’entraînement sur un petit fichier (6 Mo, à peu près 1500 exemples) est assez coûteux mais réalisable sur des machines personnelles. Sur des machines de calcul classiques (personnelles), cela prends en moyenne 6 à 10 minutes, nécessite 2-2.5 Go de mémoire.

Cependant, ce nombre d’exemples est loin d’être optimal pour entraîner le modèle vers les performances souhaitées. Lorsque la taille du jeu de données augmente, on arrive rapidement aux limites de la RAM. Par ailleurs, la possibilité d’entraîner de manière incrémentale, c’est-à-dire découper le fichier d’entraînement en plusieurs fichiers de faibles tailles n’est pas envisageable: RASA NLU n’offre pas la possibilité de réentraîner un modèle à partir d’un modèle entraîné.

Cette particularité pose aussi un point critique pour la suite pour Flybot: lorsque le chatbot sera étendu à d’autres domaines ou applications (réservations d’hôtels, de taxi, etc), il faudra entraîner de nouveau un modèle avec le jeu de données initial auquel seront ajoutés des exemplaires liés au nouveau domaine.

De plus, au déploiement, le modèle entraîné occupe beaucoup d’espace mémoire (bien plus que 512 Mo).

Ainsi, nous estimons qu’il serait intéressant financièrement pour le client d’investir dans une infrastructure de calcul performant. **Une étude quantitative portant sur le prix d’une telle infrastructure, les éventuels coûts liés à celle-ci, et les prix nécessaires pour louer une puissance de calcul et une comparaison de ces éléments est nécessaire avant de prendre une décision finale.**