Analyse de la présence des stacktrace python sur Stackoverflow.com

Grégory LEFER Alexandre FRANCOIS

02/12/2014

Table des matières

1 Introduction

Lors du développement de logiciel, il est fréquent de voir apparaitre des bugs, due à de nombreux facteurs tel que l'expérience du développeur ou la maturité d'un framework. C'est dans ce but que de nombreux sites proposent leurs services afin de permettrent aux développeurs de pouvoir partager leurs connaissances. Il existe également des sites spécialement conçus afin de pouvoir exposer les problèmes rencontrés lors d'un développement comme par exemple stackoverflow.com.

Une stack trace (ou trace d'appels en français) est un affichage de la pile d'éxécution à un instant précis de l'éxecution d'un programme. Bien que pouvant être affichée à n'importe quel moment par le développeur, cette dernière est le plus souvent utilisée lors de la phase de débuggage. C'est en général ce qui est affiché dans le terminal par l'interpréteur lorsqu'un logiciel rencontre une erreur grave. De nombreux développeurs utilisent celle-ci pour découvrir l'origine de l'erreur.

C'est sur ce constat que nous allons réalisés des statistiques concernant les pratiques des utilisateurs du site stackoverflow.com. Nous allons nous concentrer sur la présence des stack traces dans leurs questions et leur influence sur les réponses apportées (quantité, pertinence, résolution..).

```
<Posts
  Id
  PostTypeId (1=Question, 2=Answer)
  AcceptedAnswerId (only present if PostTypeId is 1)
  ParentID (only present if PostTypeId is 2)
  Score
  Body
  Tags
/>
```

FIGURE 1 – Structure du fichier Posts.xml minimifié

```
(Traceback \(most recent call last\):)
(\s+File\s+\"[^\"]+\",\s+line\s+\d+,\s+in[^\n]+\n[^\n]+\n)+
\s*\w+:[^\n]+
```

FIGURE 2 – Expression régulière reconnaissant les stacktraces Python

2 Développement

Pour réaliser cette analyse, le contenu des posts du site StackOverFlow (Posts.xml) a été utilisé. StackOverFlow est un site web pour les développeurs rencontrant divers problèmes afin de les exposer et ainsi trouver une aide extérieur. Ce fichier représente 30Go de donnée contenant l'ensemble des posts depuis la création du site jusqu'au début 2014. La librairie stAx (Streaming Api for XML) qui permet de traiter un document XML de façon simple en consommant peu de mémoire a été utilisé pour lire ce fichier.

2.1 Reconnaissance de stacktrace python

Afin de récupérer les stacktraces python un parser a été créé. Pour savoir si une stacktrace est présente une expression régulière est utilisée.

2.2 Parseur de stacktraces

Le premier dataset crée est un dataset contenant seulement des stacktraces pythons

function PARSERSTACKTRACEDATASET
  List $Posts \leftarrow read('Posts.xml')$
  for all $post \leftarrow Posts$ **do**
    List $Stacktraces \leftarrow getStacktraces(post.body)$
    for all $stacktrace \leftarrow Stacktraces$ **do**
      writer(stacktrace)
    end for
  end for
end function

FIGURE 3 – Recherche des stackstraces python

2.3 Parseur de questions contenant au moins une stacktrace

Le second dataset constitue toutes les questions contenant au moins une stacktraces dans celle-ci.

```
function PARSERQUESTIONWITHSTACKTRACEDATASET
    List Posts ← read('Posts.xml')
    List Questions
    for all post ← Posts do
        if post.postTypeId is Question then
            List Stacktraces ← getStacktraces(post.body)
            if Stacktraces is not empty then
                question = new Question(post, Stacktraces)
                Questions ← question
            end if
        else
            if post.postTypeId is Reponse then
                for all question ← Questions do
                    if question.idPost = post.idParent then
                        List Stacktraces ← getStacktraces(reponse.body)
                        reponse = new Reponse(post, Stacktraces)
                        if question.AcceptedAnswerId = reponse.idPost then
                            question.AcceptedAnswer ← reponse
                        else
                            question.reponses ← reponse
                        end if
                    end if
                end for
            end if
        end if
    end for
    writer(Questions)
end function
```

FIGURE 4 – Recherche des questions contenant une stackstraces python

2.4 Parseur de questions ne contenant pas de stacktraces

Ce dernier dataset regroupe toutes les questions pythons sans stacktrace afin de d'avoir des données de comparaison avec le deuxieme dataset

2.5 Résumé

3 datasets sont donc disponible, la raison de la création de plusieur dataset par rapport à un seul est que la génération d'un dataset demande un temps de l'ordre de 25min à 1h, la creation d'un dataset plus important aurait donc été plus long, cette solution a été adapté afin de pouvoir exploiter un dataset pendant la création d'un autre.

3 Exploitation des datasets

Cette section présente les differentes manières utilisées pour l'exploitation de ces datasets

```
function PARSERQUESTIONWITHOUTSTACKTRACEDATASET
    List Posts ← read('Posts.xml')
    Map<Int,Question> Questions
    for all post ← Posts do
        if post.postTypeId is Question AND post.Tags contains "python" then
            List Stacktraces ← getStacktraces(post.body)
            if Stacktraces is empty then
                question = new Question(post)
                Questions ←< post.idPost >< question >
            end if
        else
            if post.postTypeId is Reponse then
                tmp ← Questions.get(post.parentId)
                if tmp exist then
                    List Stacktraces ← getStacktraces(reponse.body)
                    reponse = new Reponse(post, Stacktraces)
                    if question.AcceptedAnswerId = reponse.idPost then
                        question.AcceptedAnswer ← reponse
                    else
                        question.reponses ← reponse
                    end if
                end if
            end if
        end if
    end for
    writer(Questions)
end function
```

FIGURE 5 – Recherche des questions ne contenant pas de stackstraces python

3.1 Buckets, baseX, XPath

Le dataset contenant l'ensemble des stacktraces à été diviser en bucket en fonction du type d'erreur ou exception survenue. Les datasets contenant les questions et leurs réponses associées ont été exploité grâce au logiciel baseX qui est un système de gestion de base de données XML native et légère et est spécialisé dans le stockage, le requêtage et la visualisation de larges documents et collections de documents XML. Pour effectuer nos differentes requêtes le langage XPath à été utilisé. c'est un langage pour localiser une portion d'un document XML simple d'emploi.

```
#Nombre de questions
    //Question
    Resultat: 12492
#Nombre de questions contenant une reponse acceptee
    //Question[AcceptedAnswer/Score]
    Resultat: 7167
#Nombre de questions ne contenant pas de reponse  acceptee mais ayant des reponse
    //Question[not(AcceptedAnswer/Score) and Reponses//Score]
    Resultat: 3796
#Nombre de questions ne contenant pas de reponse acceptee mais ayant des reponse avec un score positif
    //Question[not(AcceptedAnswer/Score) and Reponses//Score > 0]
    Resultat: 2283
#Nombre de questions contenant une reponse acceptee mais ayant des reponse avec un score plus eleve
    //Question[Reponses//Score  > AcceptedAnswer/Score]
    Resultat: 713
#Nombre de reponse contenant une stacktrace
    //Question[Reponses//Stack]
    Resultat: 99
```

FIGURE 6 – Requêtes effectué sur le dataset contenant les questions avec les stacktraces

```
#Nombre de questions
    //Question
    Resultat: 345217
#Nombre de questions contenant une reponse acceptee
    //Question[AcceptedAnswer/Score]
    Resultat: 215288
#Nombre de questions ne contenant pas de reponse acceptee mais ayant des reponse
    //Question[not(AcceptedAnswer/Score) and Reponses//Score]
    Resultat: 98154
#Nombre de questions ne contenant pas de reponse acceptee mais ayant des reponse avec un score positif
    //Question[not(AcceptedAnswer/Score) and Reponses//Score > 0]
    Resultat: 62907
#Nombre de questions contenant une reponse acceptee mais ayant des reponse avec un score plus eleve
    //Question[Reponses//Score  > AcceptedAnswer/Score]
    Resultat: 29485
#Nombre de reponse contenant une stacktrace
    //Question[Reponses//Stack]
    Resultat: 626
#Nombre de question ne contenant pas de reponse
    //Question[not(Reponses/Reponse)]
    Resultat: 131175
```

FIGURE 7 – Requêtes effectué sur le dataset contenant les questions sans stacktraces