Snaking the Web

Markus Franz
mail@markus-franz.de

INTRODUCTION

- Metasearch engine
- New-class: Load every webpage returned by source search engines
- Research projects of University of Hannover and NEC (around 1996)
- Problems:
  1. Load many webpages fast
  2. Use contents fast

RESTART OF DEVELOPMENT

- Idea for solving the performance problems (2004)
- Solution: parallelization
- Restarted at in December 2004
- Supported by University of Hannover and SuMa-eV

PROGRAMMING LANGUAGE

- C/C++: too complex and hard to learn
- PHP: no thread / process management
- Perl: bad syntax
- Java: Too much code
- Python: Fast, flexbile, easy to learn and use

LOADING OD WEBPAGES

- Asyncronous sockets
- Threads
- Processes

```
for url in urls_all[:]:
        pid = os.fork()
        if pid == 0:
                # do something with url
                pass
```

⇒ Processes AND asyncronous sockets

DOCUMENT CONTENTS

- Regular expressions

```
re.compile('<title>(.*?)</title>', re.I)
re.compile('(.{200})<a (.*?)href="(.+?)"(.*?)>(.+?)<\/a>(.{200})', re.I)
```

- Tried htmllib, but don‘t like it
- Process communication: temporary files
- Easily implement changes for spam filters

FUTURE WORK

- Bringing Metager2‘s Python code to other search engines
- Create an own index
- Deal with Java or C APIs

SUMMARY

- Easily write code -> good for continuous improvement (spam filtering)
- Very good methods to save search time (threads, asyncore, fork)
- Nice modules for text processing

=> Python helps to do impossible things