국민내비 김기사와 Python

- 데이터 분석을 위한 동적 웹 이미지 생성

강 대 성
daesung7.kang@gmail.com

강 대 성 ?

- 8년의 대기업 생활에 머리가 망가짐.

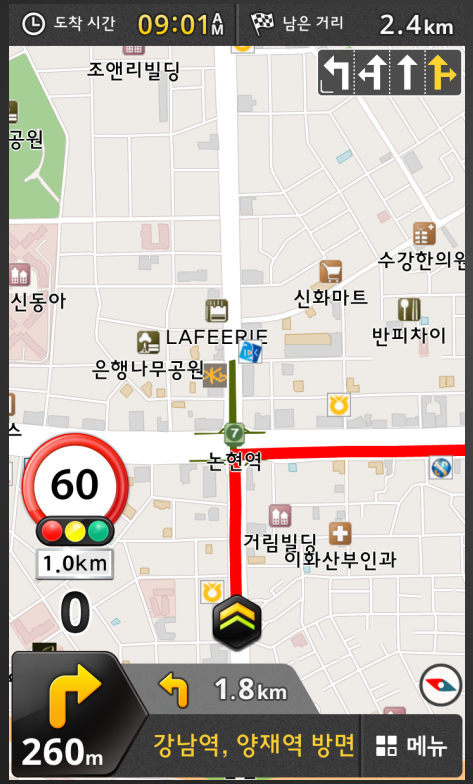
- 현)록앤올 근무 :
  김기사 데이터 분석

- 쓸데 없는 생각에 빠져 있기

- 아이스하키를 다양하게 즐기기(Play, Watch)

  Asia League Ice Hockey 2014.9월 시작! http://www.alhockey.com

국민내비 김기사?

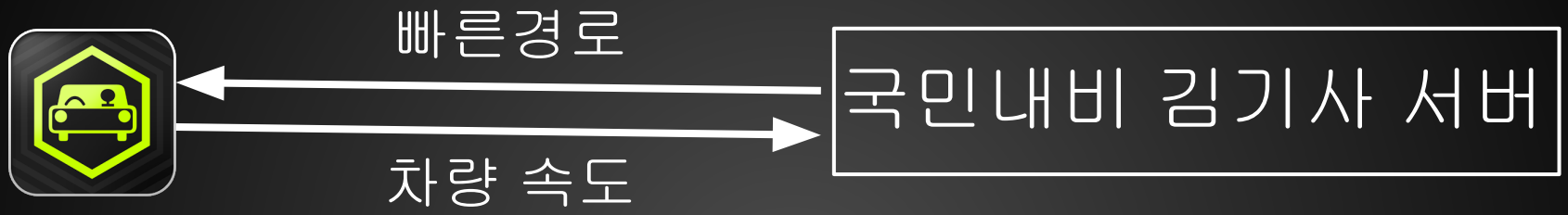


실시간 교통정보를 반영한
내비게이션 앱

Android, iOS, HTML5

가입자 : 약 800만
길안내 : 약 7500만건/월

실시간 교통정보?



앱과 서버간 데이터 교류

- 서버로부터 최적 경로와 도로의 소통 상태 전송 받음.
  (실시간 정보에 기반한 최적의 경로 안내)
- 주행 중인 도로의 소통 상태를 서버에 전송
  (자연스럽게 많은 데이터가 쌓임)

고객의 불만

* 25분 걸린다고 했는데 도착해보니 35분.
* 급격한 도로 상황 변화 시 부정확

미션 : 도착 예정시간을 개선!

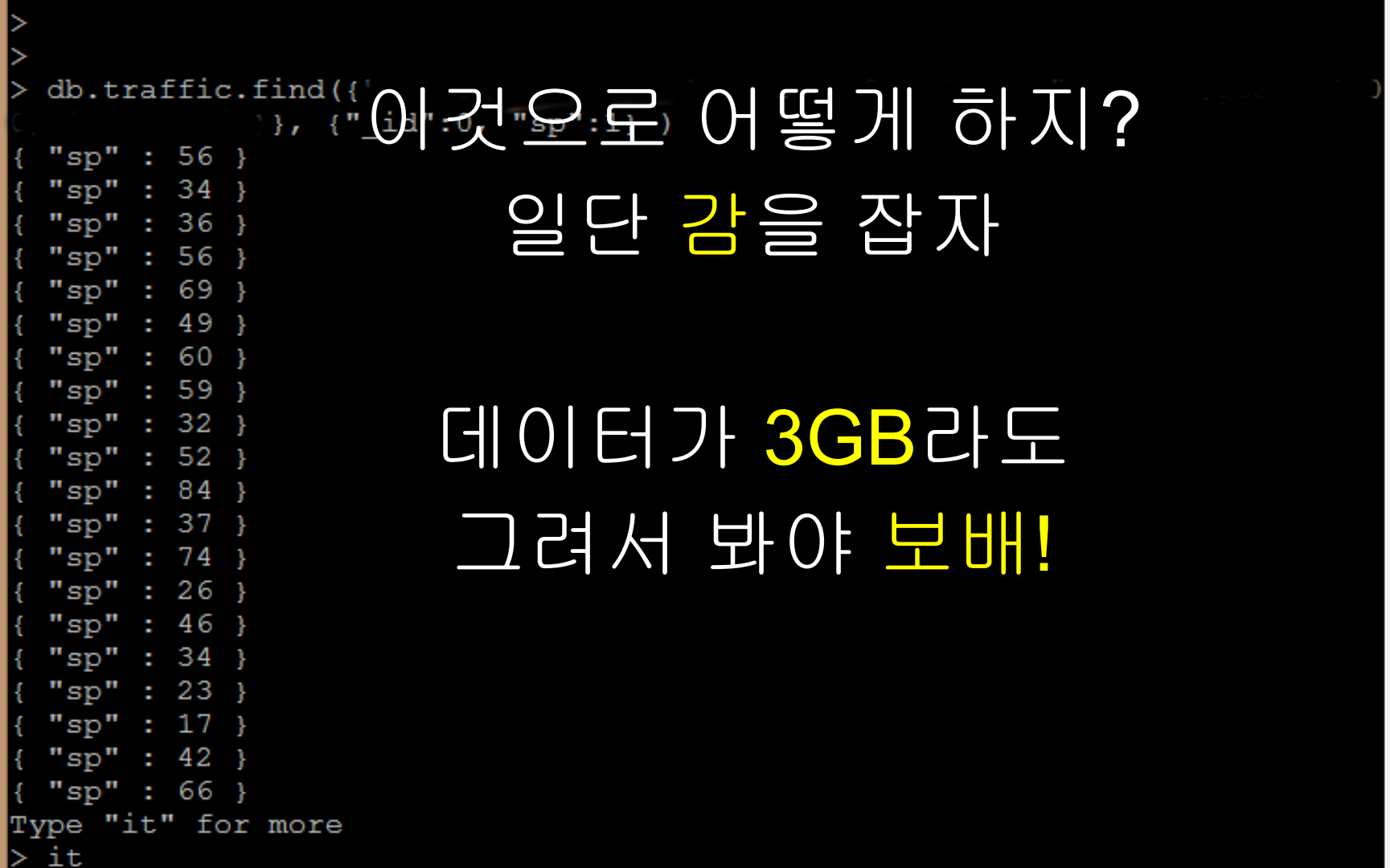
많은 데이터 처리를 위해 팀장의 권유로...

데이터 분석 시작!!

```
[hostname ~]$ mongo
MongoDB shell version: 2.4.9
connecting to: test
> use t
switched to db t
> db.traffic.find({.....})
```

db.traffic.find({.....})

```
> db.traffic.find({
                }, {"_id":0, "sp":1} )
{ "sp" : 56 }
{ "sp" : 34 }
{ "sp" : 36 }
{ "sp" : 56 }
{ "sp" : 69 }
{ "sp" : 49 }
{ "sp" : 60 }
{ "sp" : 59 }
{ "sp" : 32 }
{ "sp" : 52 }
{ "sp" : 84 }
{ "sp" : 37 }
{ "sp" : 74 }
{ "sp" : 26 }
{ "sp" : 46 }
{ "sp" : 34 }
{ "sp" : 23 }
{ "sp" : 17 }
{ "sp" : 42 }
{ "sp" : 66 }
Type "it" for more
> it
```



어것으로 어떻게 하지?

일단 감을 잡자

데이터가 3GB라도

그려서 봐야 보배!

그리자

~~Java : 회사의 주요 언어, 나에겐 익숙하지 않음~~

C/C++ : 10살 때부터 쓰긴 했지만…

Python : 익숙하지는 않은데…

C/C++(일단 쌀부터 씻고, 20분) vs Python(전자렌지 2분)

입력 :  mongoDB

처리 :  python™

+PIL(Python Image Library)

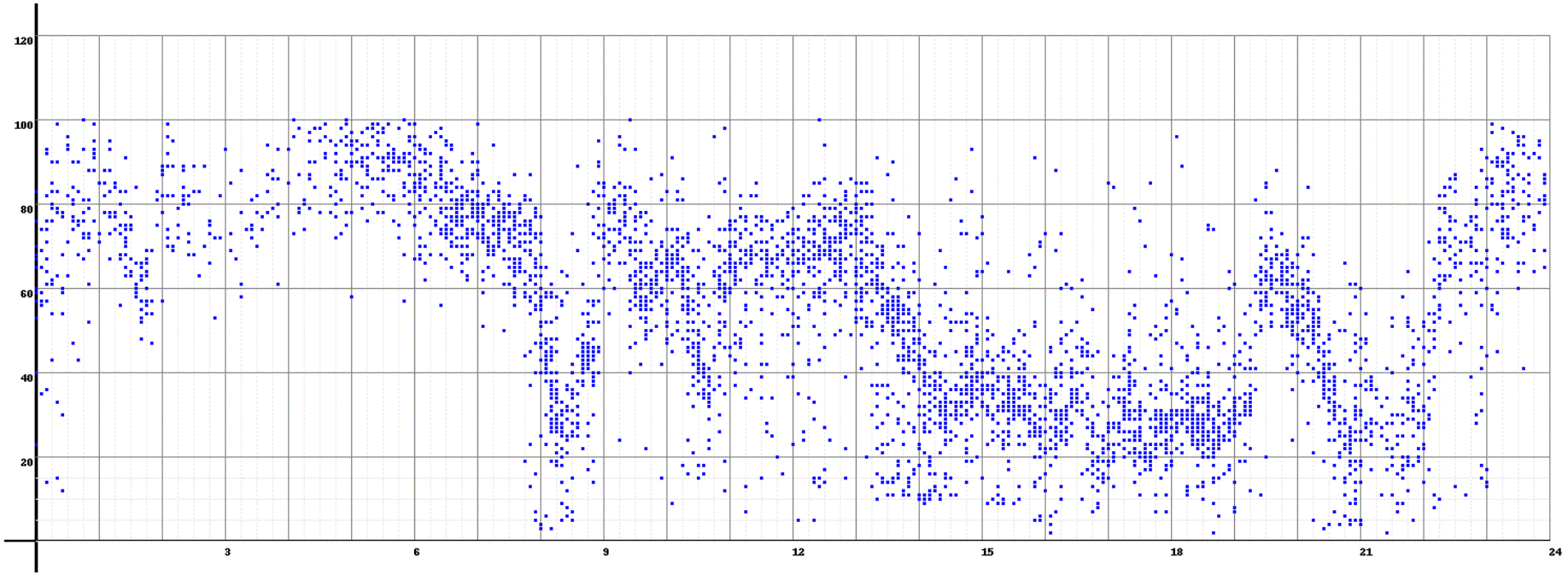
출력 : 특정 구간의 24시간을 그래프로 그리기

코드 요약

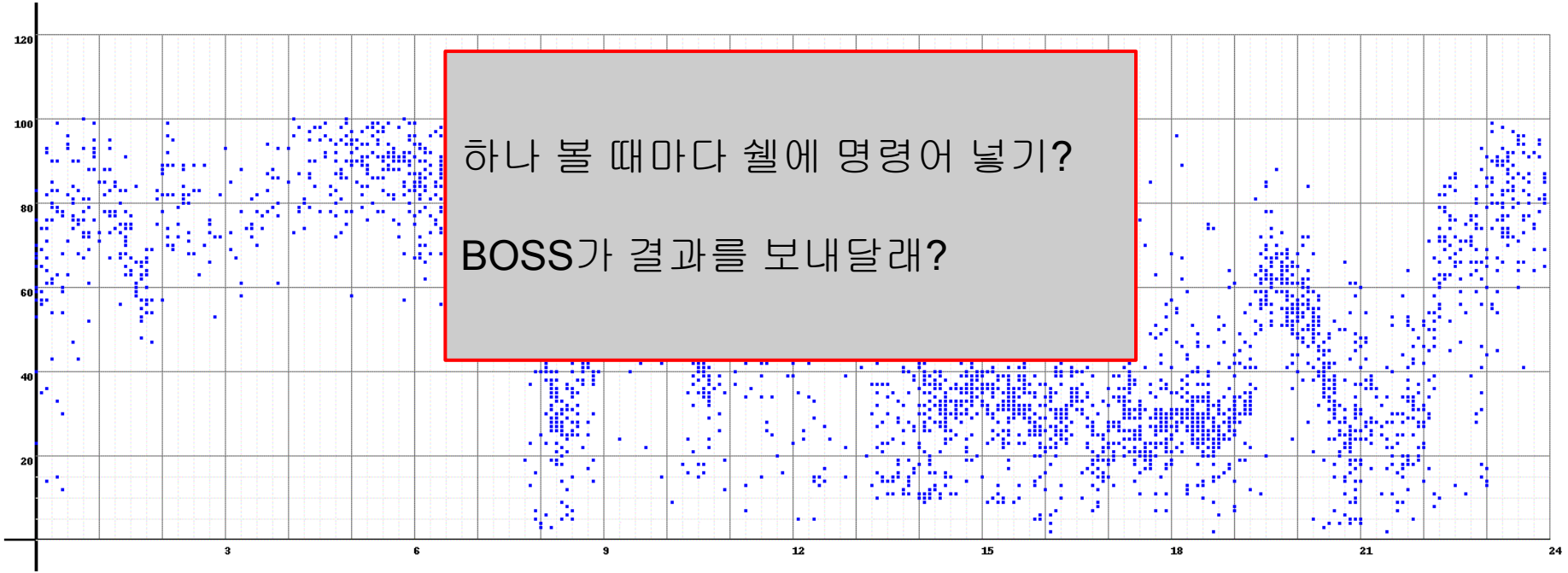
```
import Image, ImageDraw, pyMongo

img = Image.Image()
draw = ImageDraw.Draw(img)
cursor = db.find({“date”:20140829, “tid”:“12345678”})

for d in cursor:
    xy = convertxy( d[‘time’], d[‘speed’] )
     draw.point( xy, color = “blue”)
img.show()
```

보배 완성!

* 고속도로의 한 구간, 가로 : 시간, 세로 : 속도



보배를 더 좋게!


120
100
80
60
40
20
3
6
9
12
15
18
21
24
하나 볼 때마다 쉘에 명령어 넣기?
BOSS가 결과를 보내달래?

개선사항 : Web Browser로 보기

입력 : 

처리 : 

+PIL(Python Image Library)

+import BaseHTTPServer

+import CGIHTTPServer

출력 : HTTP를 이용한 전송

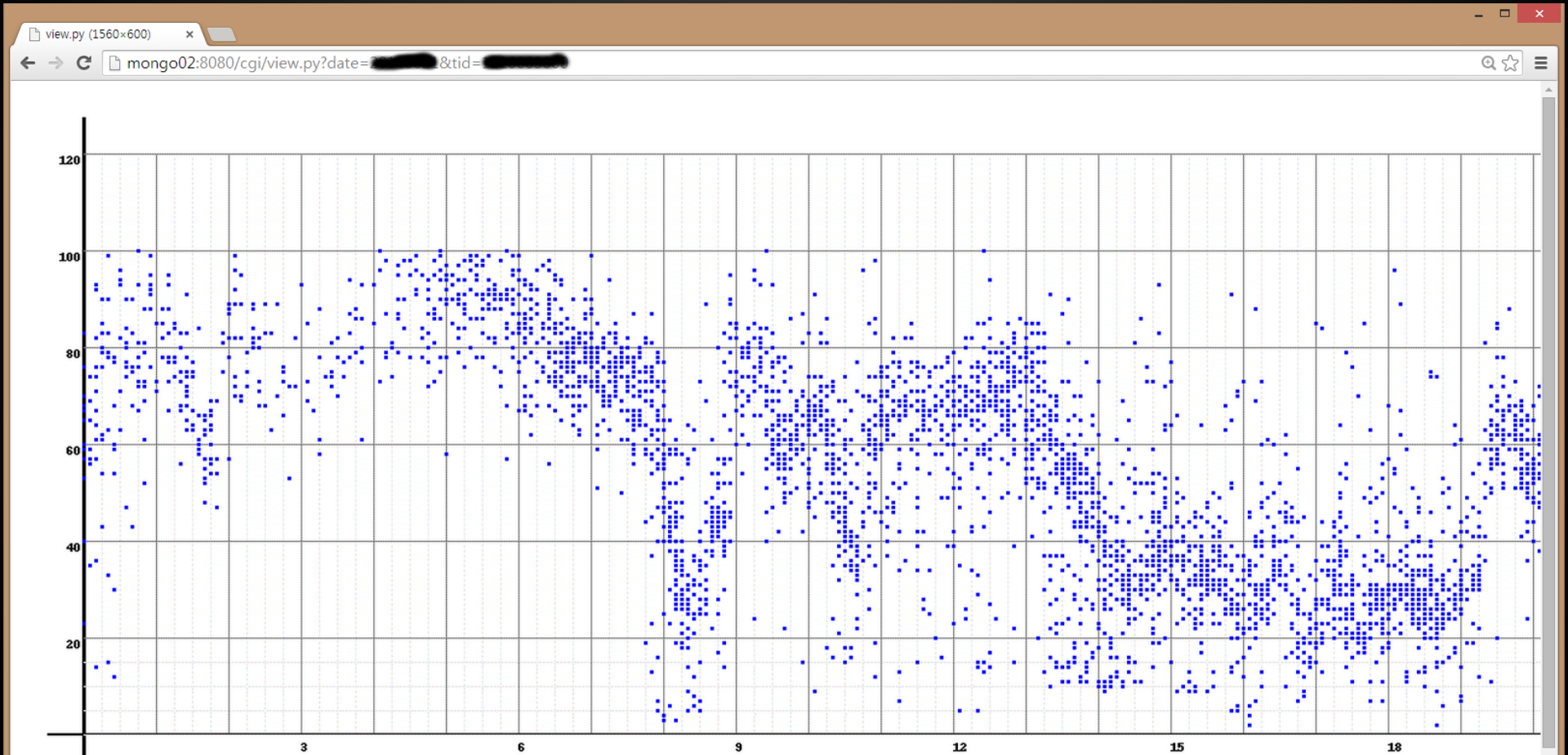
HTTP Server

```
server = BaseHTTPServer.HTTPServer
handler = CGIHTTPServer.CGIHTTPRequestHandler
server_address = ("", 8080)
handler.cgi_directories = ["/cgi"]

httpd = server(server_address, handler)
httpd.serve_forever()
```

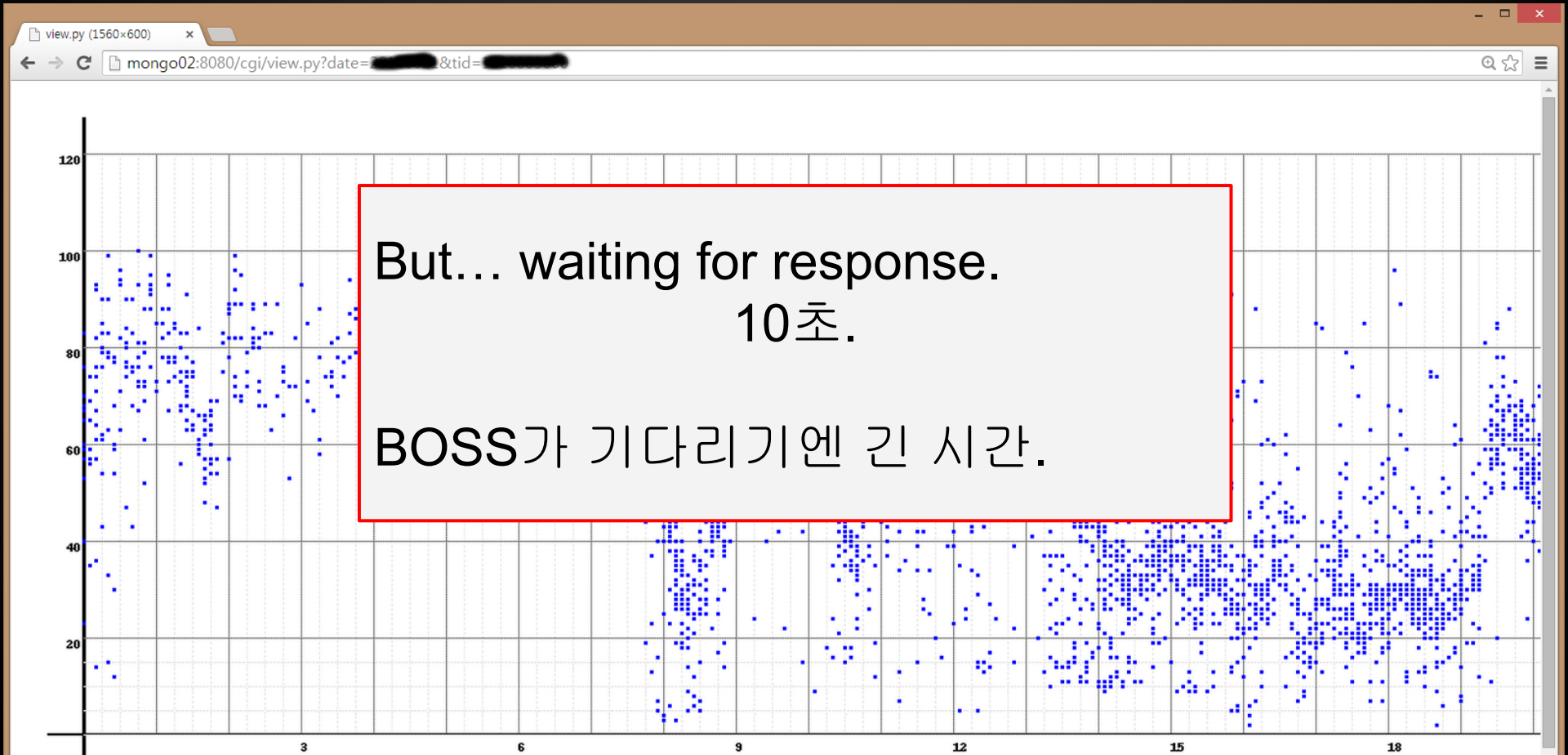
CGI

```
virtualfile = cStringIO.StringIO()

img = Image.Image()
...
img.save(virtualfile, “gif”)

data = f.read()
print "Content-type: image/gif"
print "Content-Length: " + str( len(data) )
print
print data
```

Web Browser 로 보인다.



Web Browser 로 보인다.



Cache 생성

```
cachefile = str(date)+"_"+str(tid)+".jpg"
if os.path.isfile(cachefile) and allowcache: # 파일이 있고 allowcache가 True
    # java Script로 onload가 되었을 때 allowcache=False 옵션으로 CGI호출
    data = "<html><body><img src="' +cachefile+ "' onload=....."  > </body></html>"
    print "Content-type: text/html"
    print "Content-Length: " + str(len(data))
    print
    print data
else:    # 파일이 없거나 allowcache가 False
    .......
    im.save(f, "gif")
    im.save(cachefile, quality=50)
```

Cache 생성

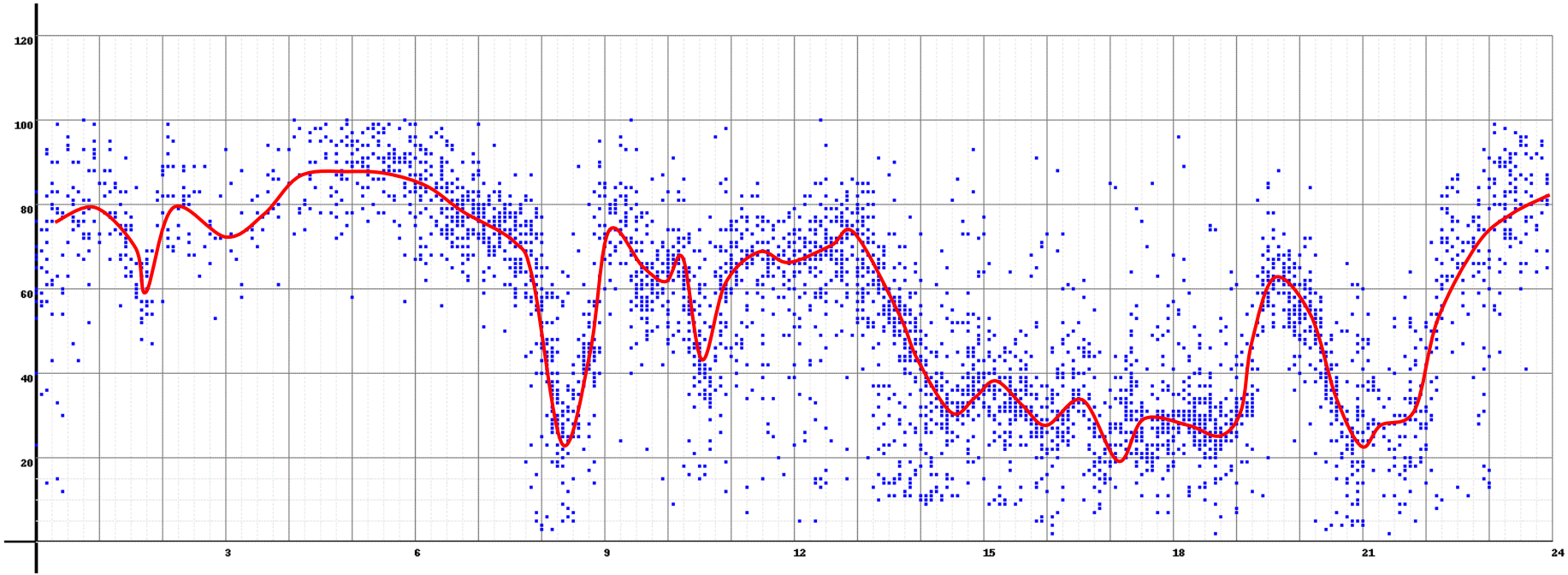
```
cachefile = str(date)+"_"+str(tid)+".jpg"
if os.path.isfile(cachefile) and allowcache: # 파일이 있고 ... ue
    # java Script로 ... 호출
    data = ... y></html>"
    print "C...
    print "C...
    print ...
    print dat...
else:    # 파일...
    .....
    im.save(f, "gif")
    im.save(cachefile, quality=50)
```

IT WORKS!

캐시가 생성된 데이터는 바로 보여주고,
약 10초 후 갱신된 내용을 보여줌

BOSS는 더 기다리지 않음.

분석할 때도 시각화! (데이터 & 결과)
다른 사람에게 설명할 때도 시각화!



미션결과 ? : Python을 이용한 그래프와,
Python으로 작성한 알고리즘과,
Python으로 작성한 시뮬레이션.

시뮬레이션(과거 1000개 경로) 만족!

현재 운행 테스트 중.

감사합니다.

빠른 개발은 **Python,**
빠른 길 안내는 국민내비 김기사