

過剰な負荷からPCを守る 通知システムの開発

NE20-1076H 薬師神 虎治郎

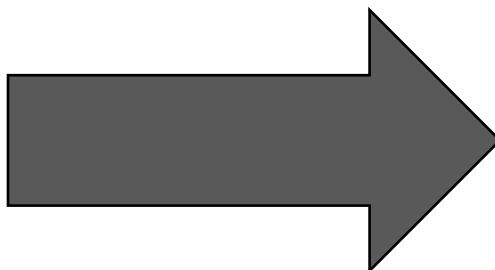
PCを利用していて思うこと、、、

MacBookは高いから長く使って欲しいなあ

負荷がかかりすぎてないかなあ、



負荷がかかっている時



負荷がかかる（PCが熱い）ケース

- ・たくさんのソフトを同時に使用している？
- ・負荷のかかるソフトの使用？



今回選んだケース

- ・たくさんのソフトを同時に使用している？
- ・負荷のかかるソフトの使用？



考えられた課題

パソコンに負荷をかけたくない！

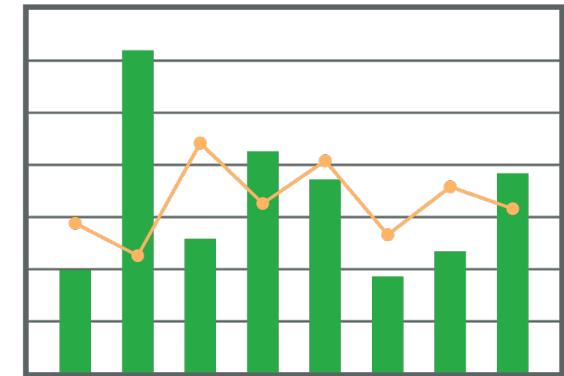
長く使って欲しいから！

課題と想定していることが本当か実験

ThingSpeakを利用して、

ソフトの重なるごとに

CPUの温度を上がるかを確認する。



ペルソナ（対象者）

専修大学情報ネットワーク学部生（筆者）

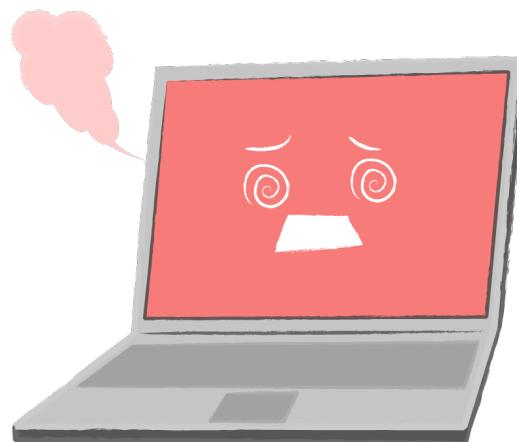
多くのソフトを同時に利用することがある
(筆者の観点から)



CPUの温度の目安

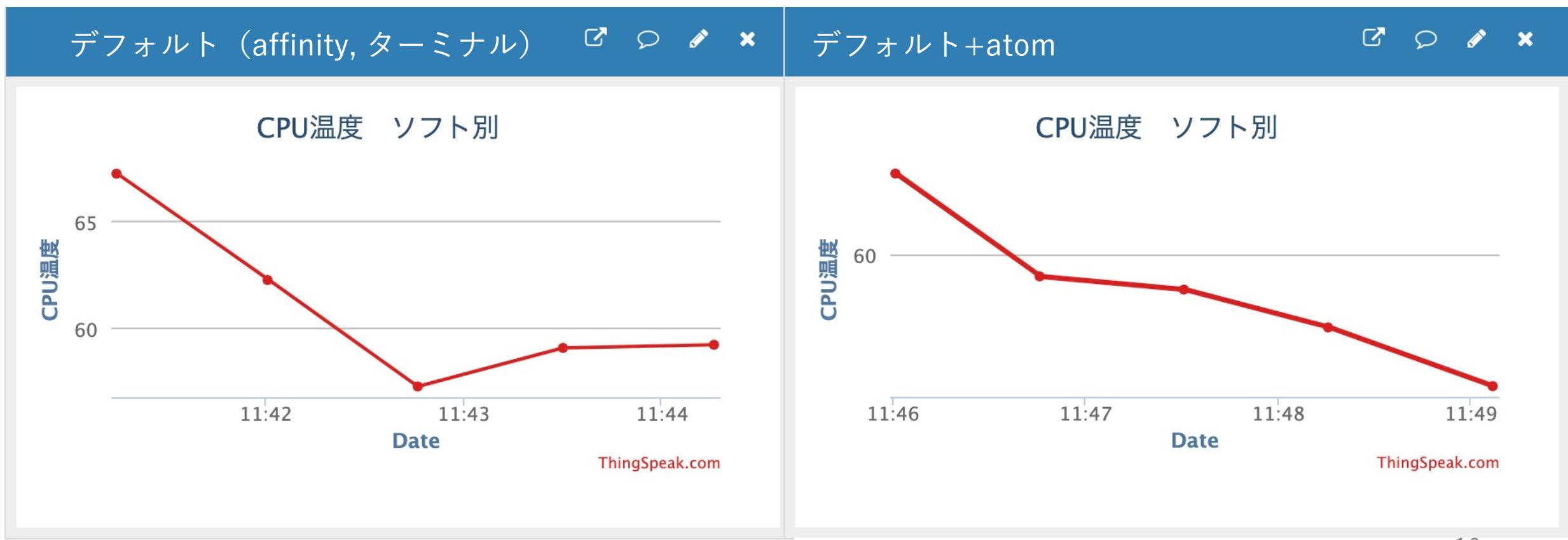
通常利用時 : 50度前後

高負荷時 : 80度超え

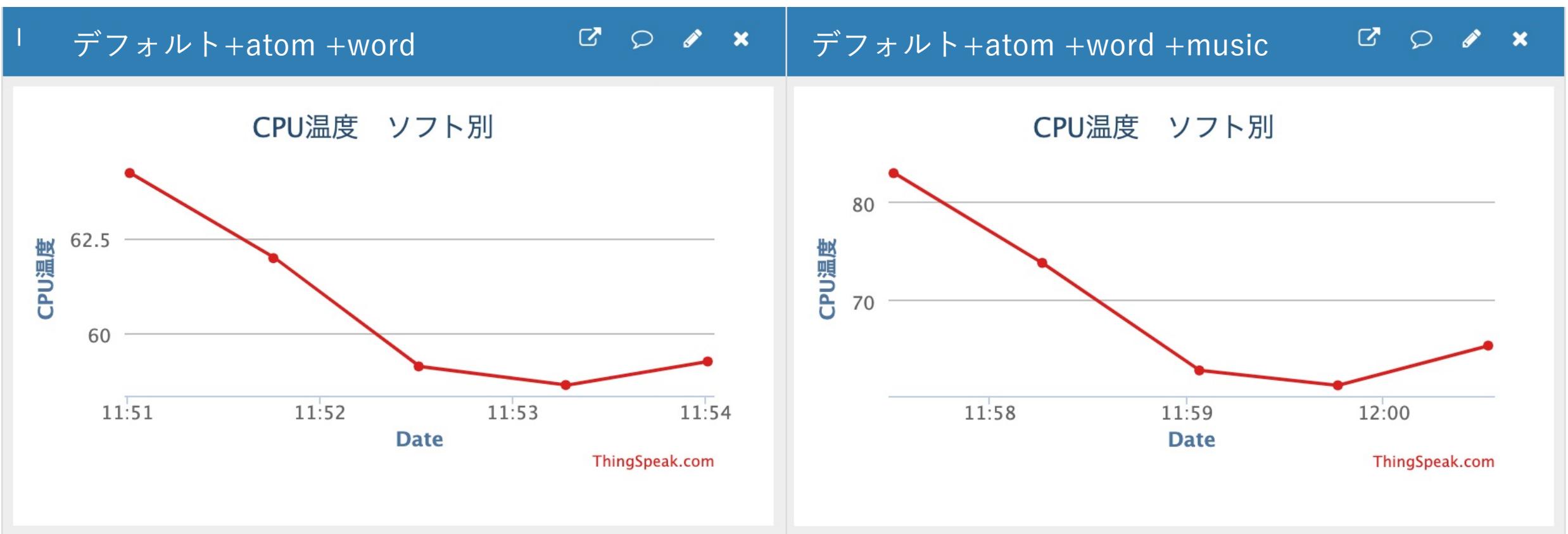


※諸説あり

同時に作動しているソフトの数で比較



同時に作動しているソフトの数で比較



データから分かること

- ・ソフトが重なる数とCPU温度は関係がない
→重なる数で負荷はかかりにくい？
- ・ソフトの起動時にCPU温度が上昇している
→起動時に負荷がかかる



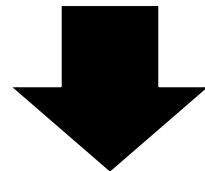
課題を解決するには

- ・一定の温度以上に達した時に通知
- ・新たにソフトを開かないようにする

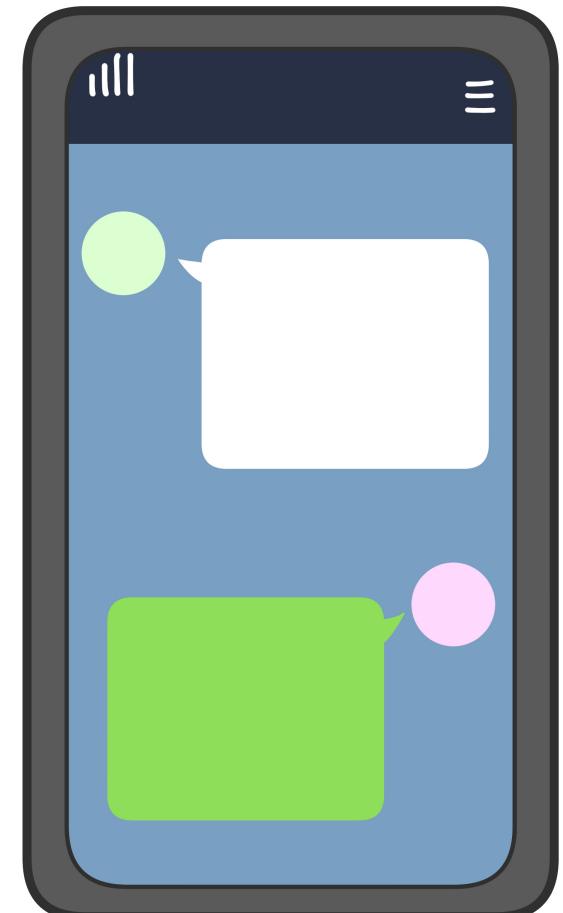
これ以上開くと、
負荷がかかる！

考案したサービス

LINE Notifyを利用して、
CPUの温度が一定に達したら、通知する



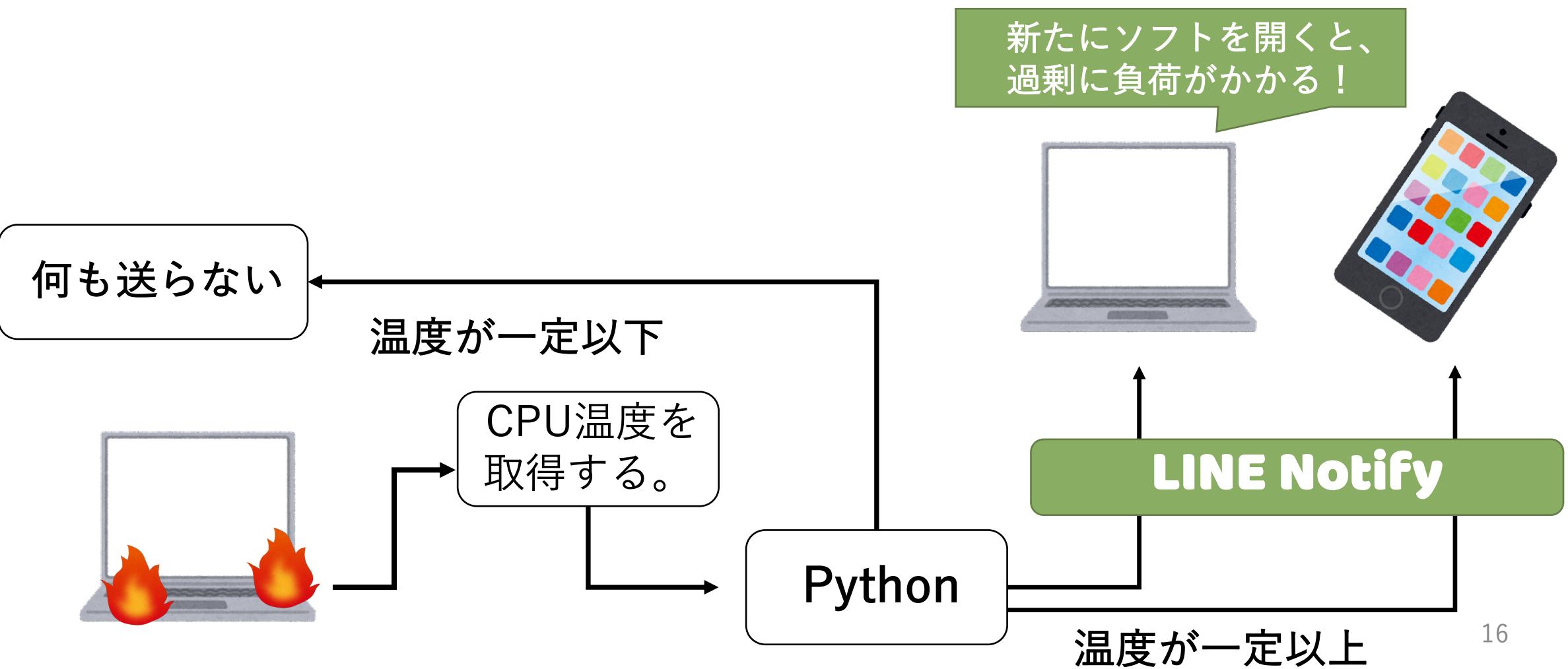
過剰な負荷をかけずに済むことができる。



必要なもの

- PC (CPU温度を測る機器)
- Python
- LINE Notify
- LINEを受け取る機器 (PC, スマホなど)

システム構造図



結果（完成することができなかった）

Pythonのコードを組み立てることができなかった



CPU温度とLINEの連携を
うまく行うことができなかった。

Pythonコード（実行できなかった）

```
cpu_notify.py          notify_test.py *
```

```
1 import requests
2
3 url = 'https://notify-api.line.me/api/notify'#LINE NotifyのAPIのURL
4 token = '個人情報' #自分のアクセストークン
5 ms = "新たなソフトを開くと負担が過剰にかかることがあります。"#送信する通知内容
6
7 def line(message,url,token):
8     post_data = {'message': message}
9     headers = {'Authorization': 'Bearer ' + token}
10    #送信する
11    res = requests.post(url,
12                          data=post_data,
13                          headers=headers)
14    print(res.text)#メッセージが送信されたかどうかの確認
15
16 while True:
17     now=dt.( 'cpu_temps')
18     dt = getCpuTempFromFile(data_file) #CPU温度取得
19     print(cpu_temps)
20     if print(cpu_temp) == "print >= 80":#CPU温度が80度以上の際にラインが送られるようにする
21         line(postdate=message, date=postdate, palams=postdate )#lineを呼び出す
22         break
23     time.sleep(1)
```

温度のデータを取得する



CPUの温度が80度以上

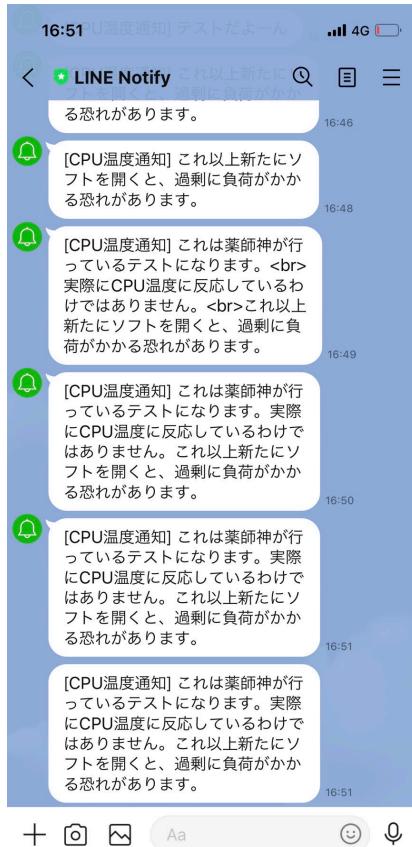


通知を送信する

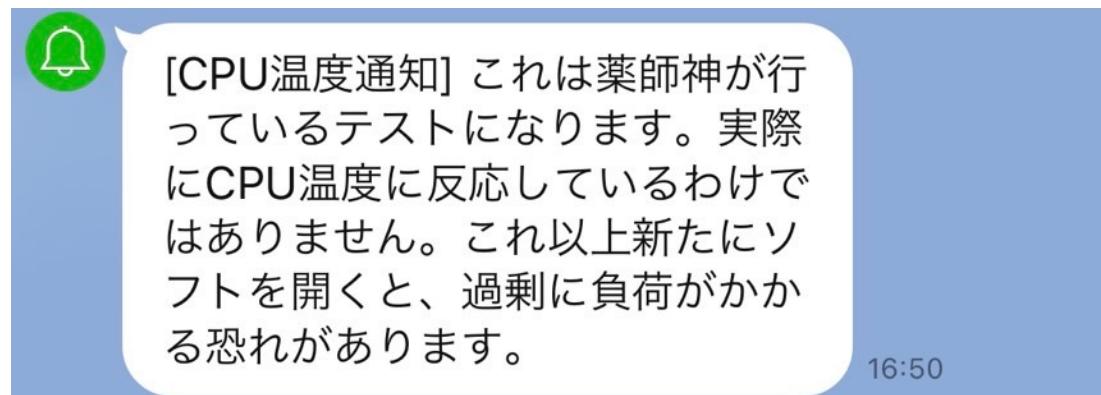
未完成だが、できしたこと

- CPU温度の取得
- LINE Notifyを利用したラインの通知

LINEを送ることはできた。（温度関係なし）



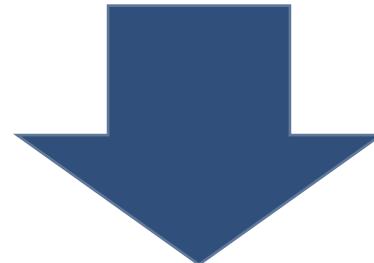
このような通知は送信できた。



完成するためには

CPU温度の取得

この連携が



実装の鍵

LINE Notifyを利用したラインの通知

= 成功（実装可能）

改善点

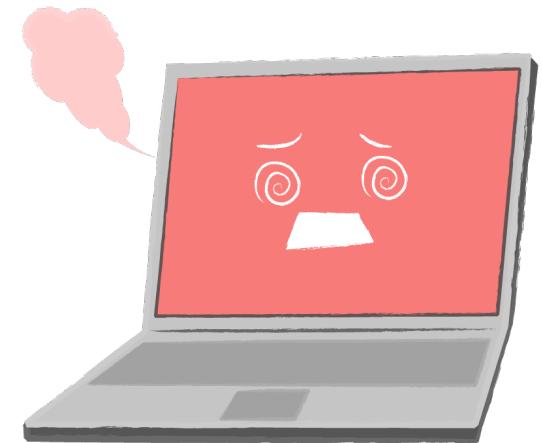
- ・ Pythonコード（連携部分）
- ・ LINE Notifyの連携の理解不足



まとめ

課題：パソコンに負荷がかかるないように、LINEを送る

できしたこと：CPU温度の取得、LINE通知



できなかったこと：PythonによるCPU温度とLINEの連携

最終制作を通じて得られたこと

- ThingSpeakの理解
- Pythonの理解
- ソフトとCPU温度の関係性
- LINE Notifyの活用性



最終制作の感想

非常に楽しかった

見えていなかつたものが見えた