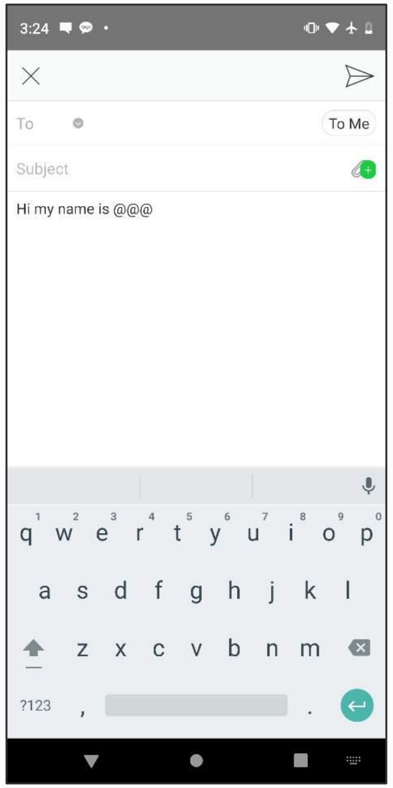
# EditText View Optimization

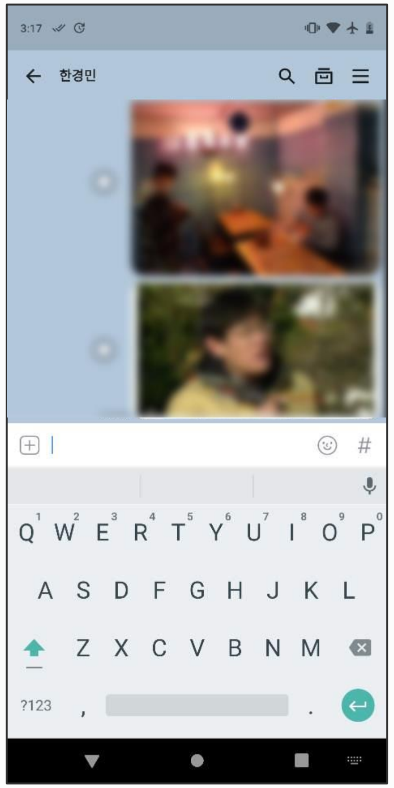
*Kyeong-Min Han*

# Introduction

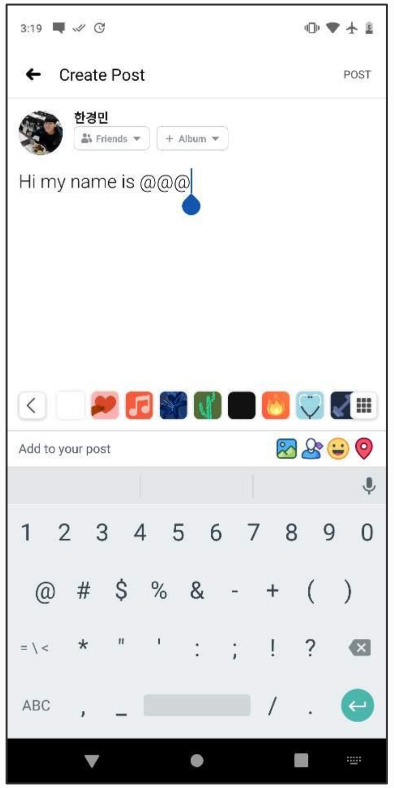
- EditText is a widget commonly used in Android.



< Naver Mail >



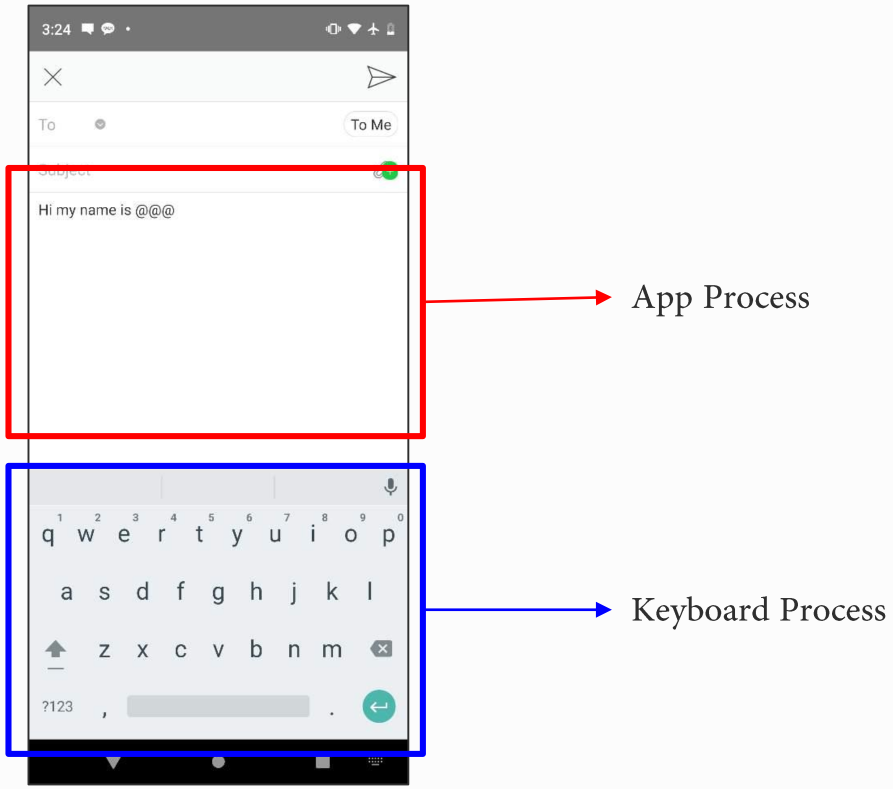
< Kakao Talk >



< Facebook >

# Introduction

- In the typing situation, the user has two interactions with Android processes .



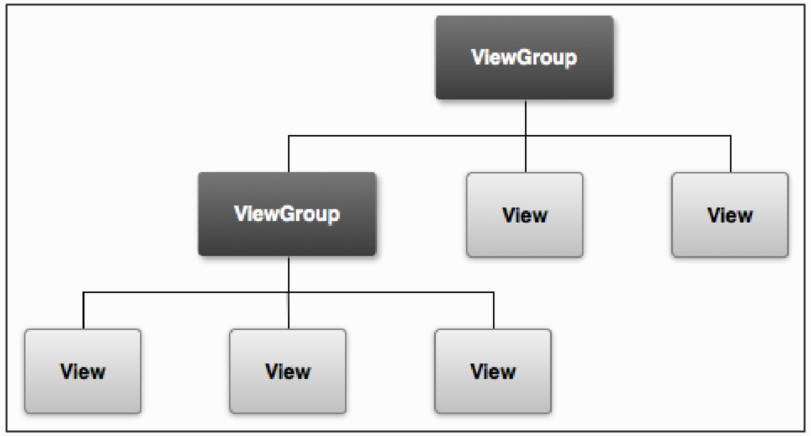
Hi my name is @@@

< Naver Mail >

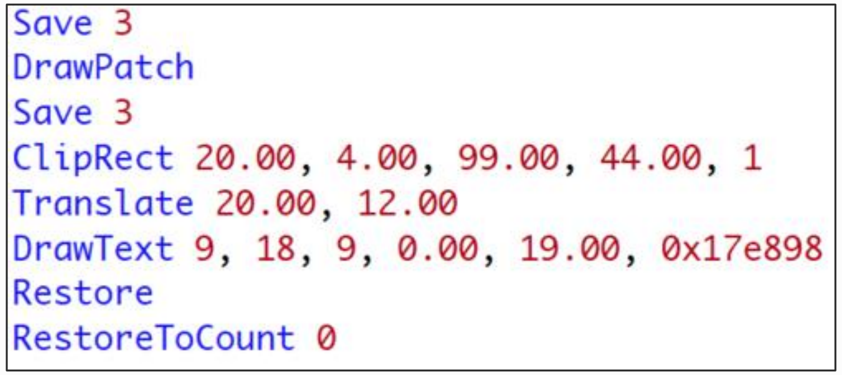
# Background

☐ ViewTree, DisplayList

< app >



< ViewTree >

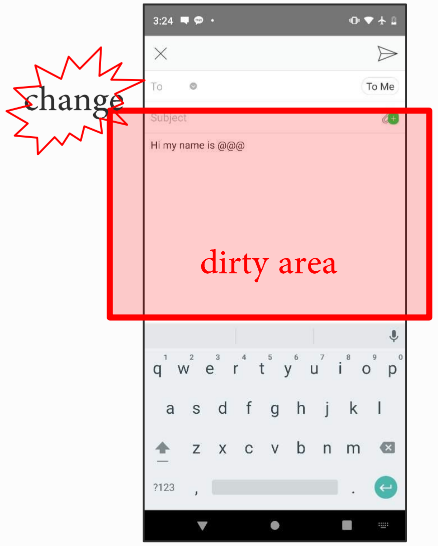


```
Save 3
DrawPatch
Save 3
ClipRect 20.00, 4.00, 99.00, 44.00, 1
Translate 20.00, 12.00
DrawText 9, 18, 9, 0.00, 19.00, 0x17e898
Restore
RestoreToCount 0
```

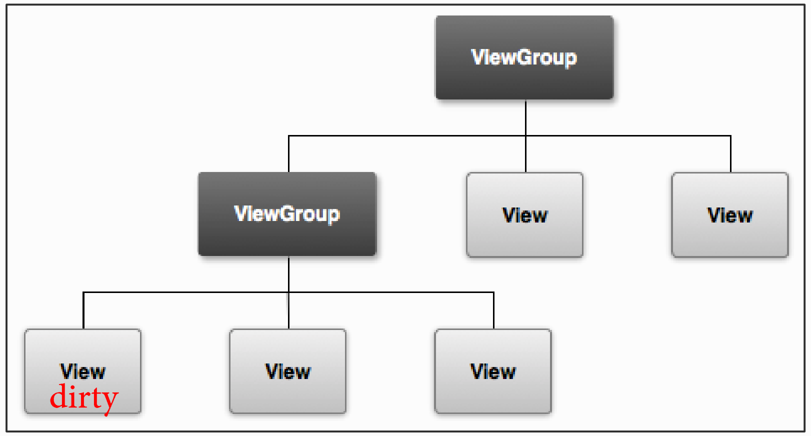
< DisplayList >

## Background

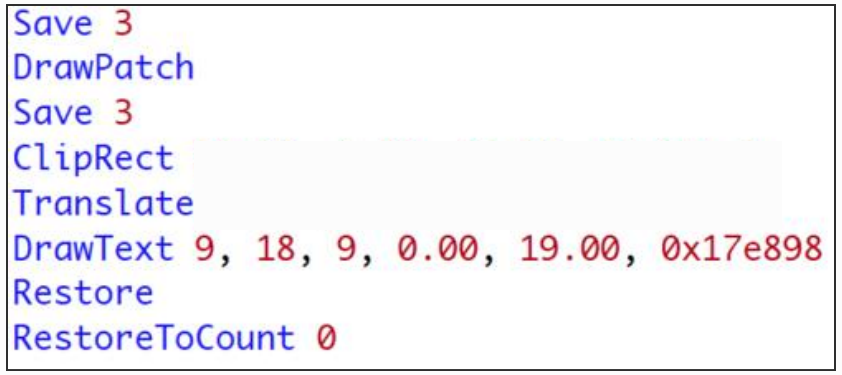
- ViewTree, DisplayList



< app >



< ViewTree >



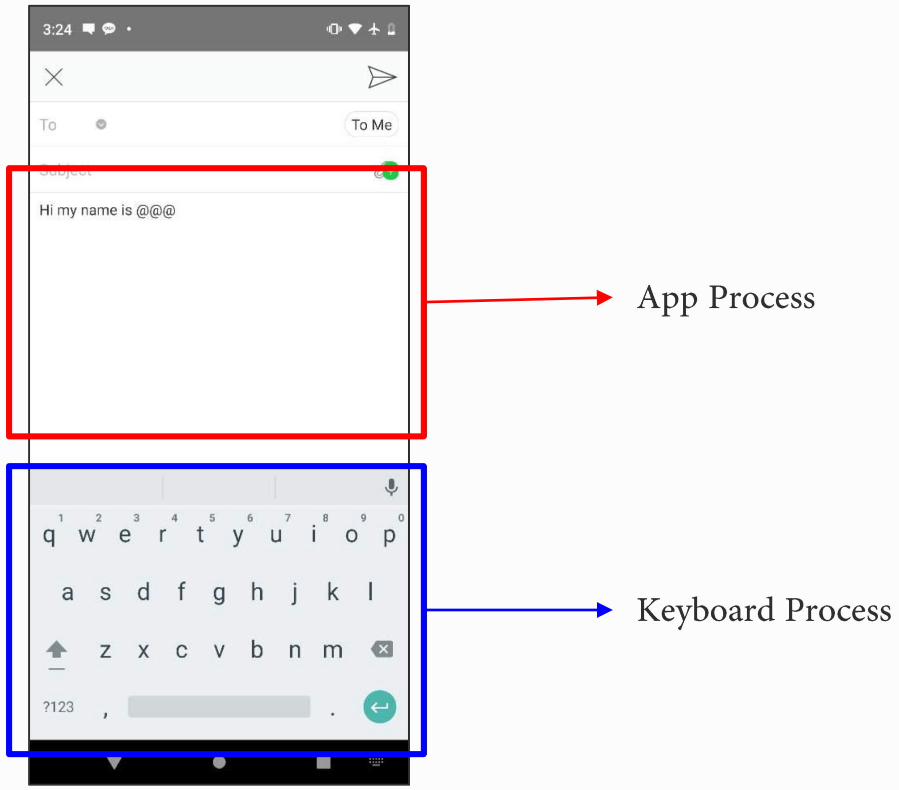
```
Save 3
DrawPatch
Save 3
ClipRect
Translate
DrawText 9, 18, 9, 0.00, 19.00, 0x17e898
Restore
RestoreToCount 0
```

< DisplayList >

## Problem

KWANGWOON UNIVERSITY

- CPU
  - Unnecessary view tree searching
  - onPreDraw (App Process)
- GPU
  - Unnecessary dirty area

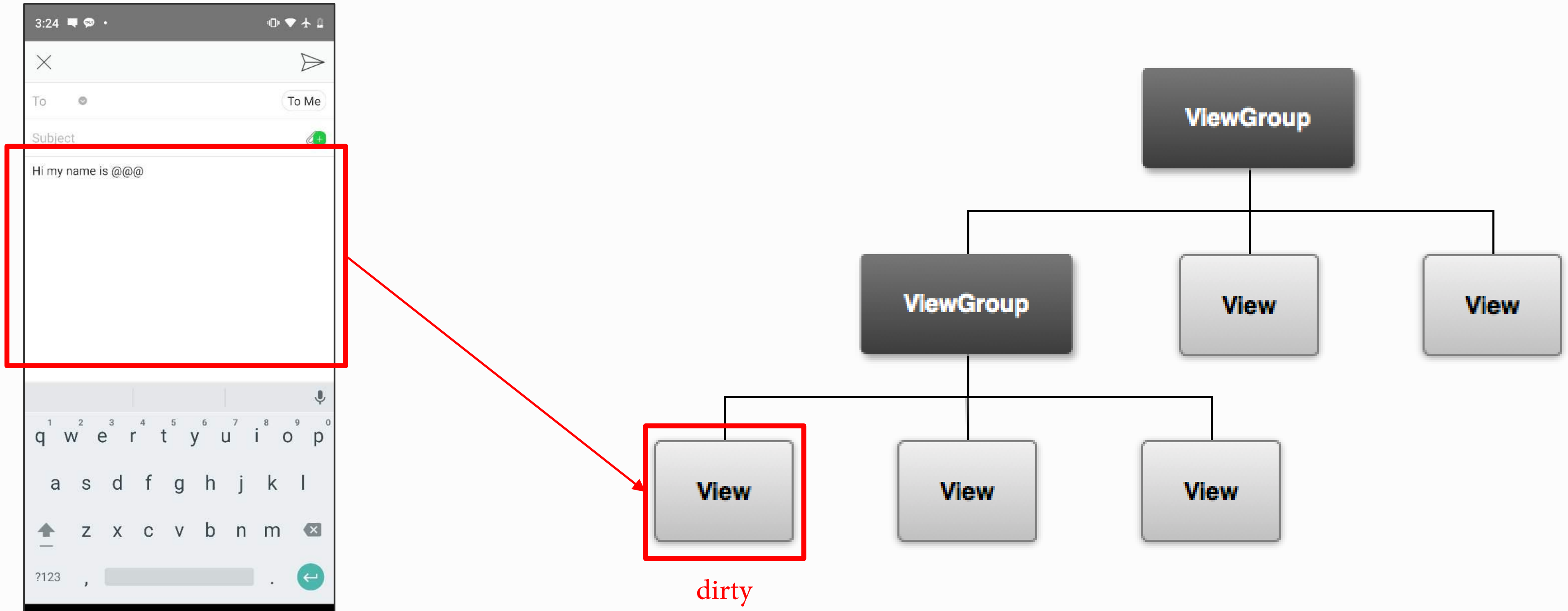


< Naver Mail >

## Problem

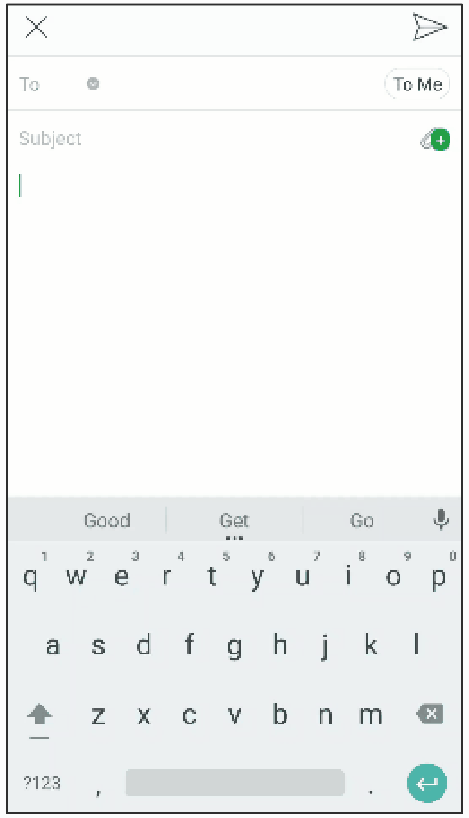
KWANGWOON UNIVERSITY

- Unnecessary view tree searching

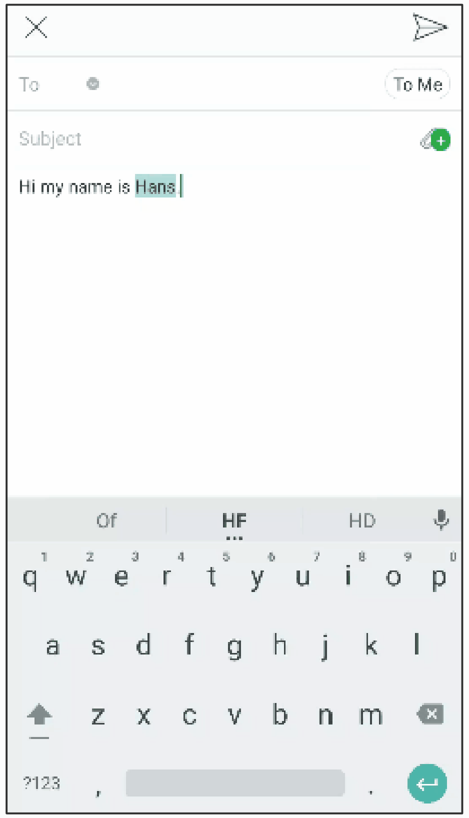


## Problem

- Unnecessary dirty area



< editText draw area >



< keyboard draw area >

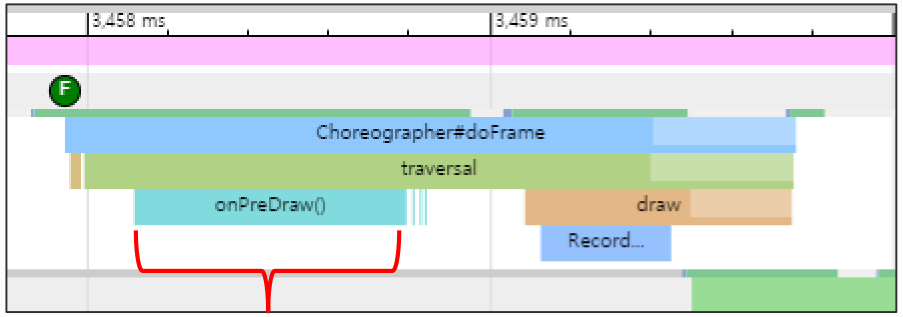
## Problem

- onPreDraw (app process)

Drawing Path

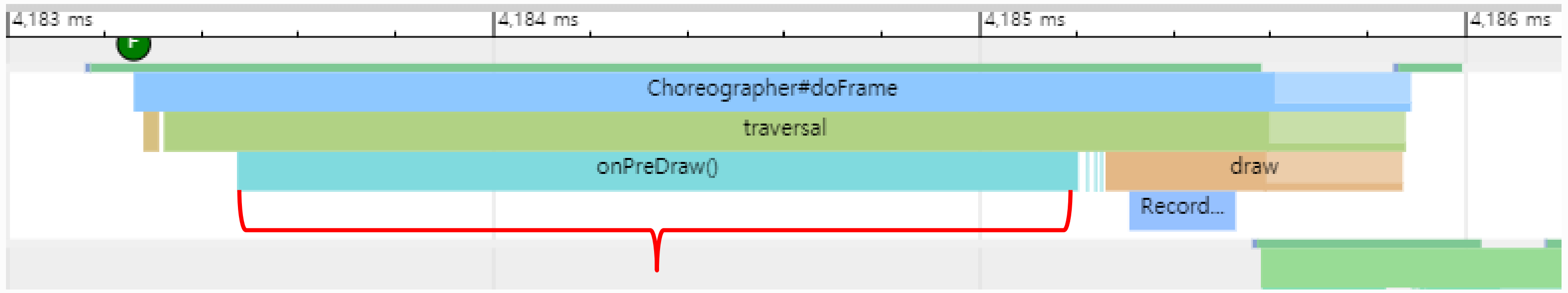
↓

```
If ( onPreDraw() || ! isViewVisible )
{
  cancel!
}
```



0.676ms

(a) When editText is empty



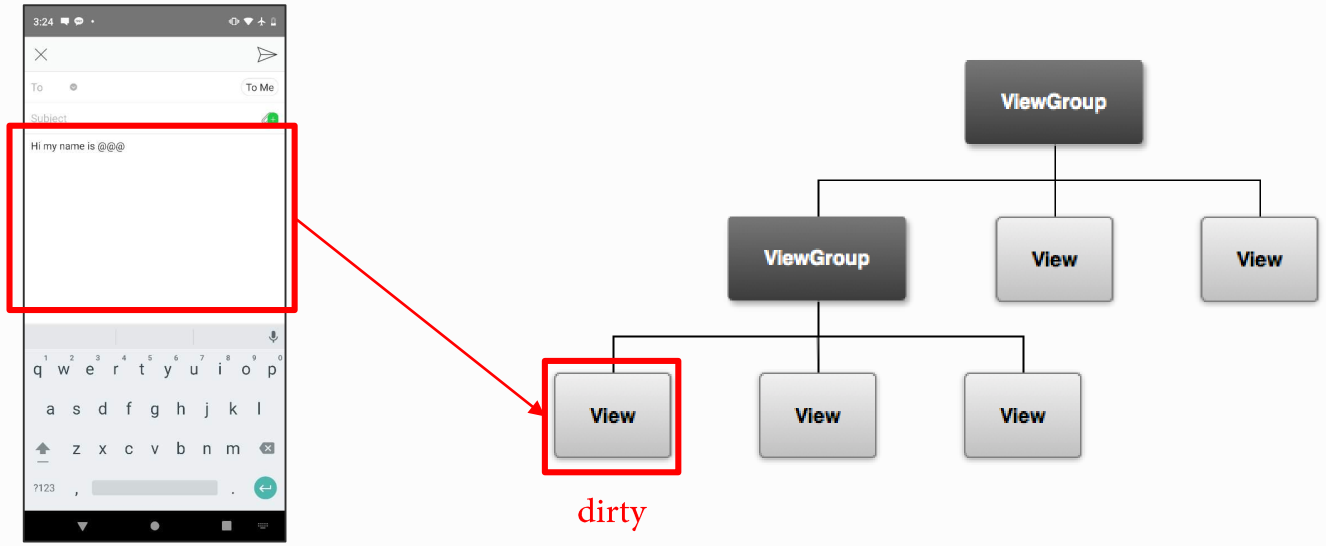
1.730ms

(b) When there are 200 characters in editText

## Solution

- Unnecessary view tree searching (App Process)

```
If ( editText is focused AND RootView is not dirty )
{
    update DisplayList(editText)
}
Else
{
    update DisplayList(RootView) //orignal code
}
```

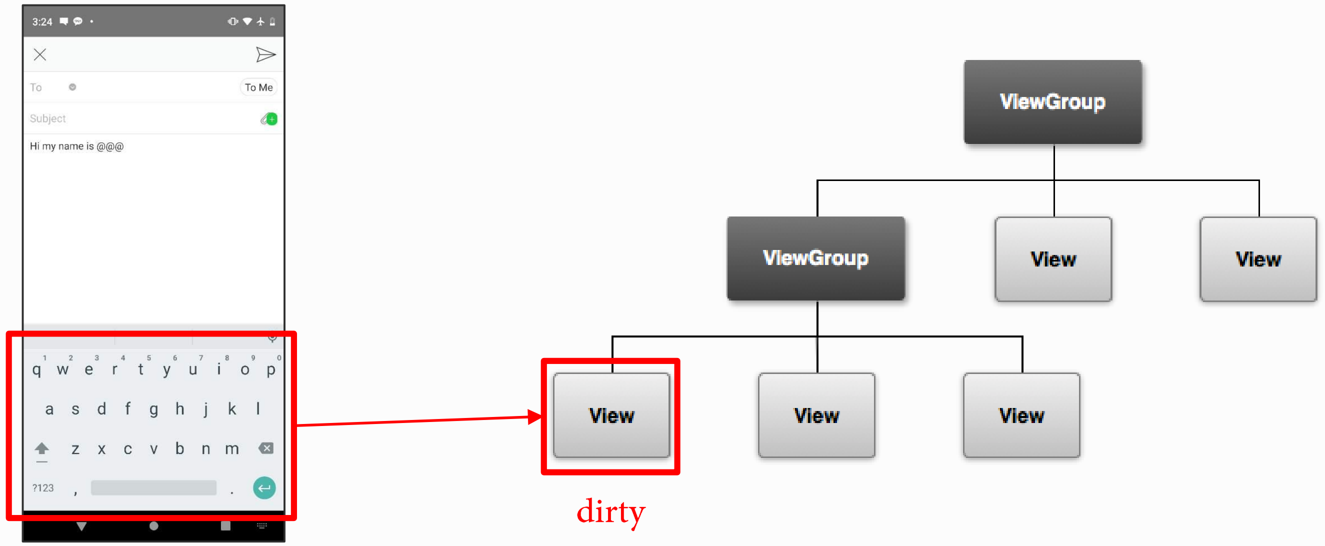


# Solution

KWANGWOON UNIVERSITY

□ Unnecessary view tree searching (Keyboard Process)

```
If ( keyboard is focused AND RootView is not dirty )
{
      update DisplayList(keyboard)
}
Else
{
     update DisplayList(RootView) //orignal code
}
```

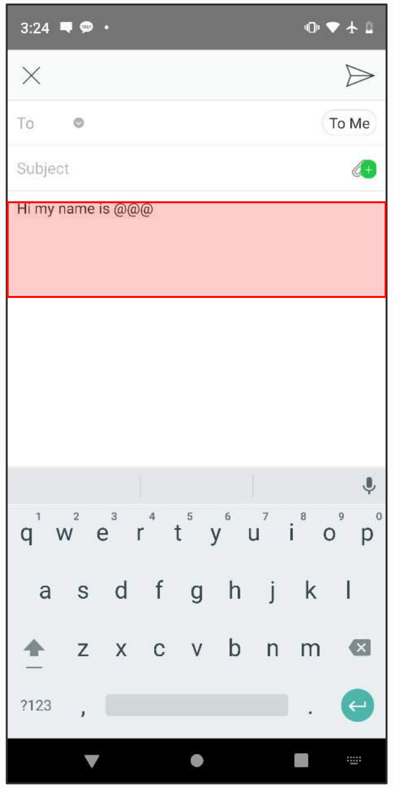


## Solution

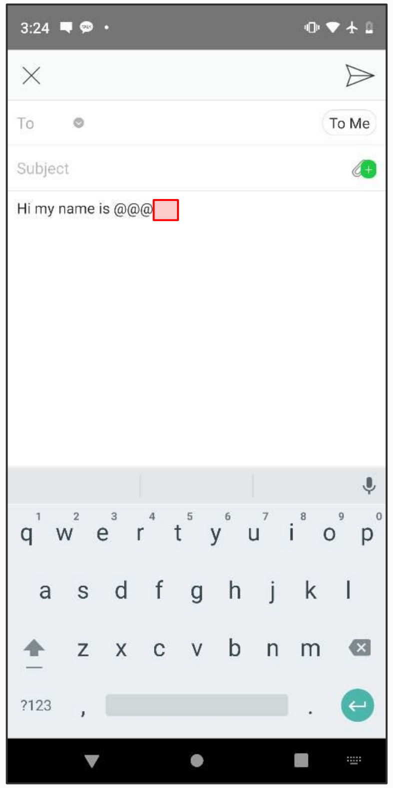
KWANGWOON UNIVERSITY

- Unnecessary dirty area (App Process)

```
Load previos Cursor x,y
Get current Cursor x,y

If ( previos x,y != current Cursor x,y )
{
    set dirty area( x, y, x+a, y+a)
    previous Cursor<=current Cursor
}
```



< original OS >

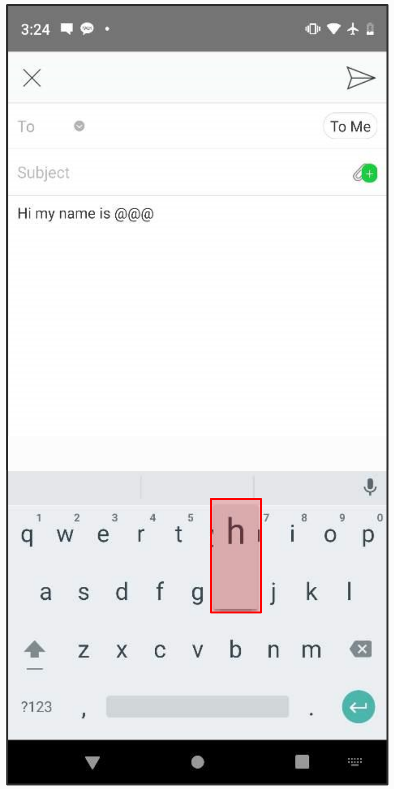
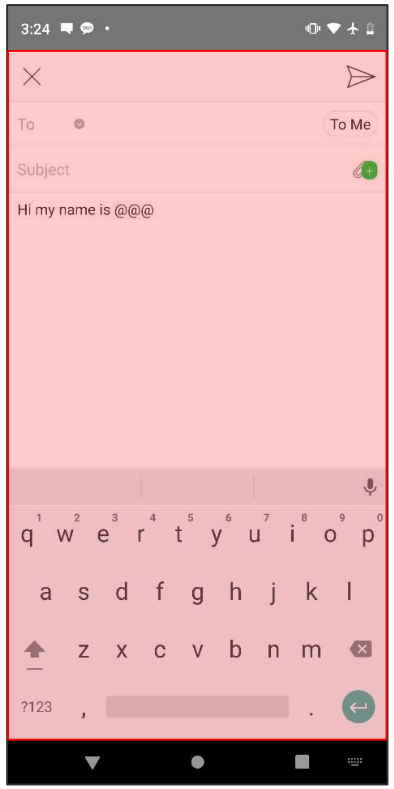


< edited OS >

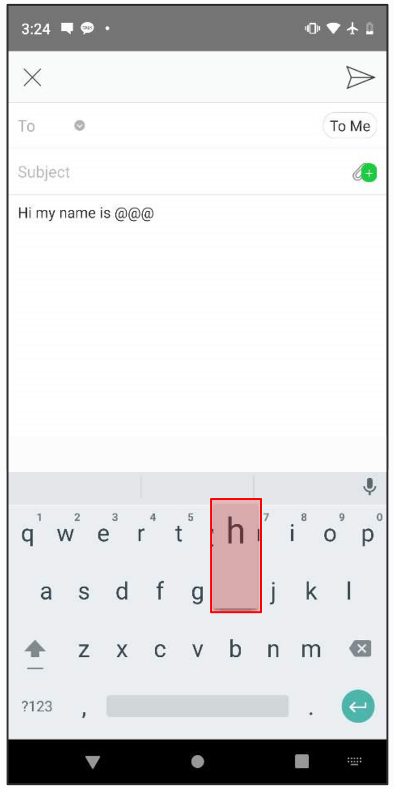
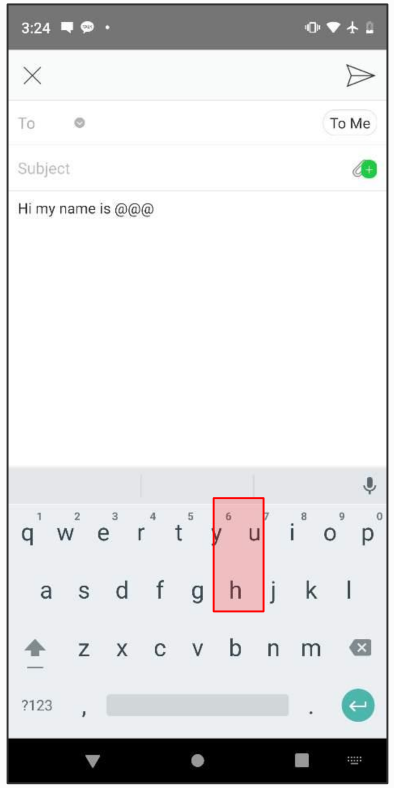
## Solution

KWANGWOON UNIVERSITY

- Unnecessary dirty area (Keyboard Process)

< original OS >

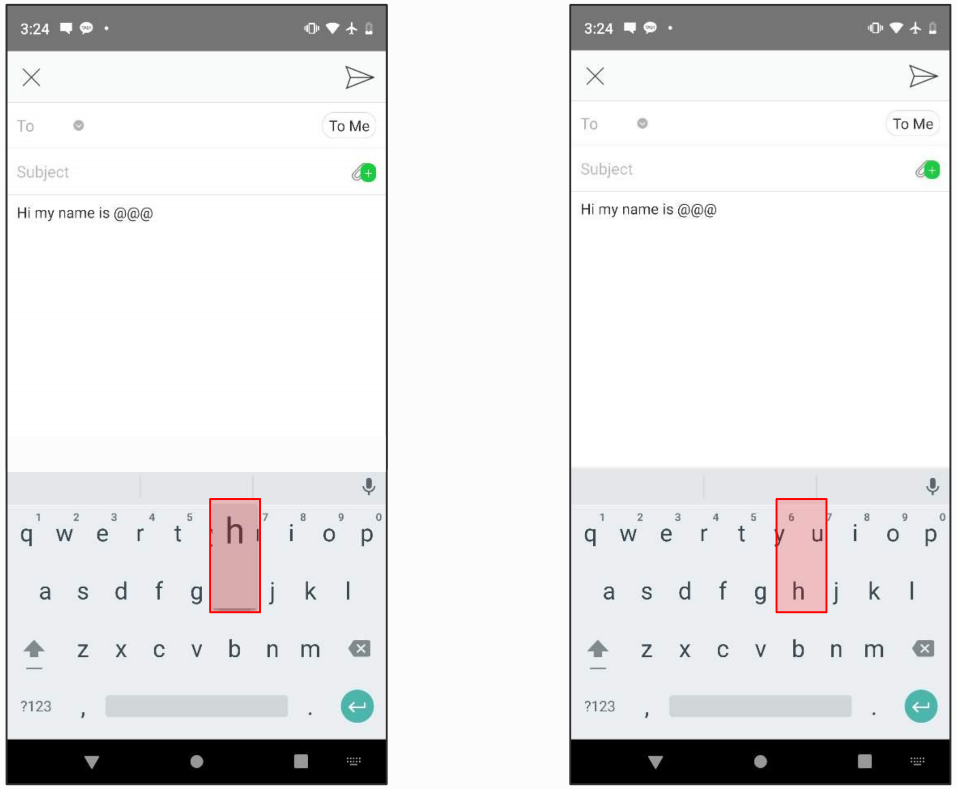



< edited OS >

## Solution

□ Unnecessary dirty area (Keyboard Process)

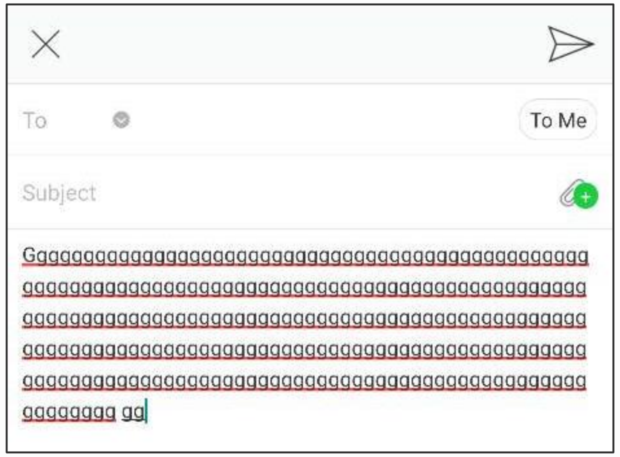
```
Get Touch point x,y
Find button location x,y by Touch point x,y

set dirty area( x, y, x+a, y+a)
```



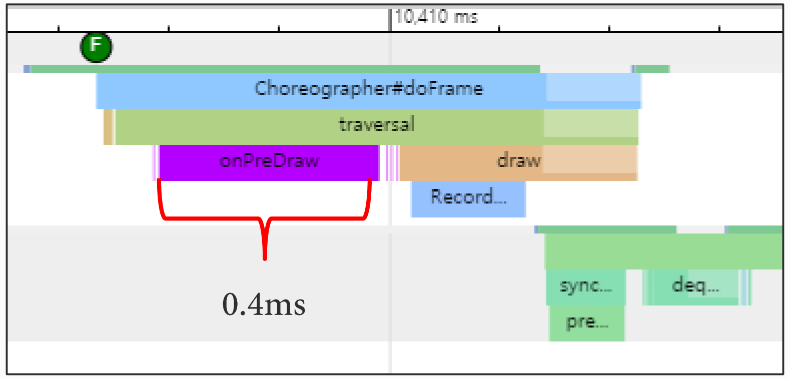
< edited OS >

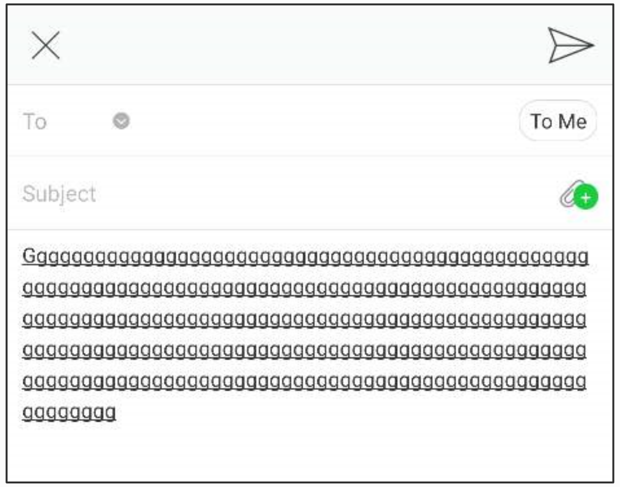
## Solution

- onPreDraw (App Process)

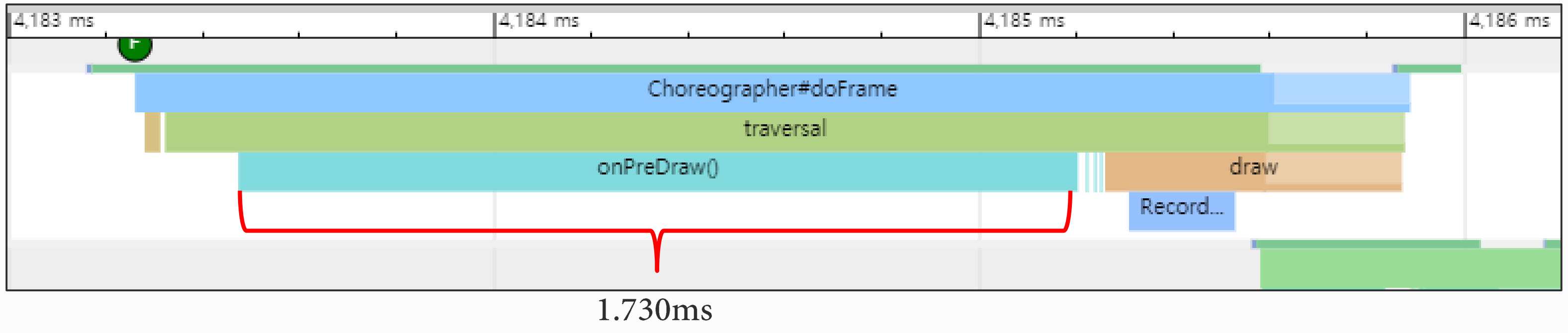

To
To Me
Subject

Gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggg gg




To
To Me
Subject

Gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggggggggggggggggggggggggggggggggggggggggggg
gggggggg

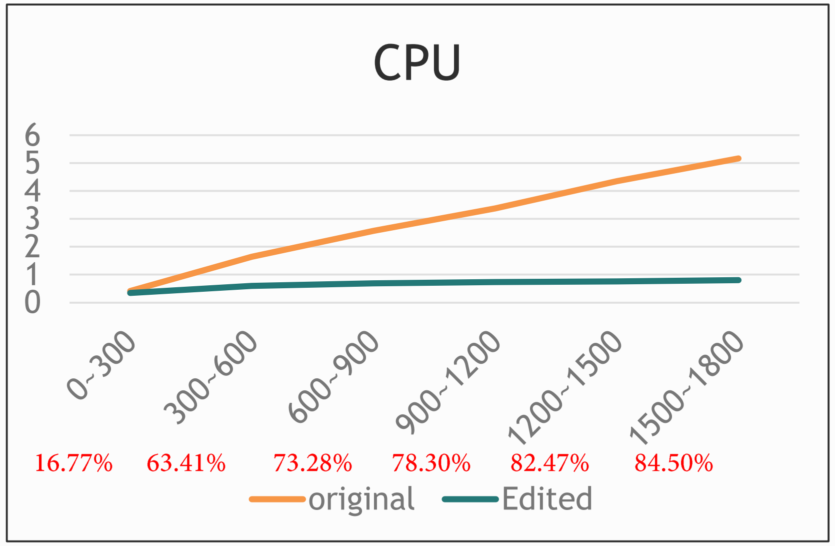


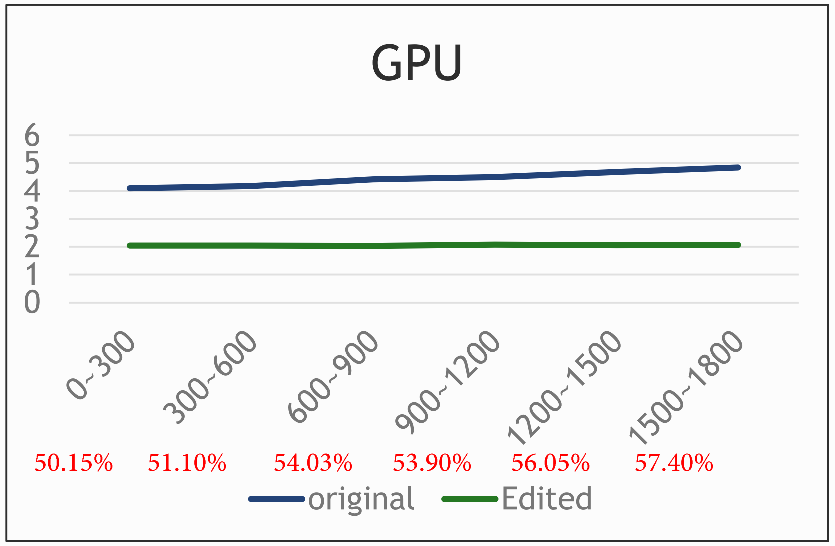
## Evaluation

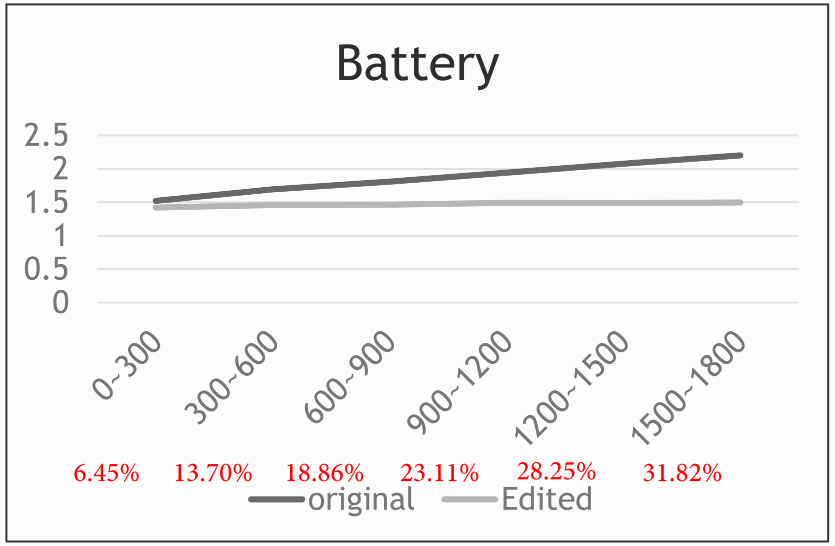
- Experiment setup
  - OS : Android API 29(Pie) releases 9.0.0.r_44
  - Hardware :  Google pixel 3XL
  - Measure Program : snapdragon Profiler
  - Target app : Naver mail
  - Typing speed : 180bpm
  - Typing time : 1min 40second ( 300 characters )
  - Screen brightness : lowest
  - Battery : 70% ~ 80%

# Evaluation

□ Naver mail







## Evaluation

| 다음 메일 | 0~300 | 300~600 | 600~900 | 900~1200 | 1200~1500 | 1500~1800 |
|---|---|---|---|---|---|---|
| CPU | 18.06% | 61.32% | 71.81% | 77.65% | 81.43% | 83.87% |
| GPU | 51.63% | 51.51% | 52.67% | 53.41% | 54.98% | 55.82% |
| Battery | 2.05% | 11.18% | 13.92% | 21.16% | 23.80% | 30.18% |

| Color memo | 0~300 | 300~600 | 600~900 | 900~1200 | 1200~1500 | 1500~1800 |
|---|---|---|---|---|---|---|
| CPU | 42.37% | 65.18% | 72.67% | 77.78% | 80.41% | 82.49% |
| GPU | 55.43% | 55.80% | 55.76% | 55.25% | 55.62% | 55.85% |
| Battery | 3.29% | 10.06% | 15.50% | 21.22% | 24.33% | 27.14% |

| 네이버 블로그 | 0~300 | 300~600 | 600~900 | 900~1200 | 1200~1500 | 1500~1800 |
|---|---|---|---|---|---|---|
| CPU | 34.65% | 65.19% | 72.02% | 76.45% | 79.20% | 81.62% |
| GPU | 54.78% | 55.99% | 54.53% | 54.76% | 54.18% | 54.05% |
| Battery | 7.78% | 12.96% | 19.56% | 24.29% | 27.34% | 30.69% |

## Evaluation

| 페이스북 | 0~300 | 300~600 | 600~900 | 900~1200 | 1200~1500 | 1500~1800 |
|---|---|---|---|---|---|---|
| CPU | -3.78% | 41.63% | 56.58% | 65.25% | 70.25% | 74.50% |
| GPU | 48.98% | 45.59% | 45.50% | 45.02% | 44.98% | 44.64% |
| Battery | 3.48% | 12.96% | 18.96% | 25.88% | 30.90% | 36.07% |

| Itranslate | 0~300 | 300~600 | 600~900 | 900~1200 | 1200~1500 | 1500~1800 |
|---|---|---|---|---|---|---|
| CPU | 14.70% | 62.02% | 72.34% | 76.72% | 80.04% | 81.98% |
| GPU | 47.46% | 42.97% | 39.57% | 38.89% | 38.90% | 38.80% |
| Battery | 4.80% | 11.15% | 17.41% | 22.06% | 24.50% | 28.79% |