

# CMPS 202

Win 2018 Final Project

Throughput: Jackon-Core  
Latency: Vanilla Music Ap

Ziye Han  
[zhan12@ucsc.edu](mailto:zhan12@ucsc.edu)

# Throughput

**Test Target: Jackson-Core**

**Github: <https://github.com/FasterXML/jackson-core>**

**Description: It is a jar plugin for JSON String serialization and deserialization**

# Latency

**Test Target: Vanilla Music App**

**Github: <https://github.com/vanilla-music/vanilla>**

**Description: It is a music player on Android**

# What Did I Run Throughput

Performance counter stats for 'java -jar MultithreadJackson.jar 1 1 200 input.txt':

36529.050710	task-clock (msec)	#	1.032 CPUs utilized	
7,354	context-switches	#	0.201 K/sec	
738	cpu-migrations	#	0.020 K/sec	
123,596	page-faults	#	0.003 M/sec	
127,846,159,328	cycles	#	3.500 GHz	(29.40%)
339,888,524,920	instructions	#	2.66 insn per cycle	(45.65%)
68,579,867,047	branches	#	1877.406 M/sec	(29.43%)
86,017,261	branch-misses	#	0.13% of all branches	(26.57%)

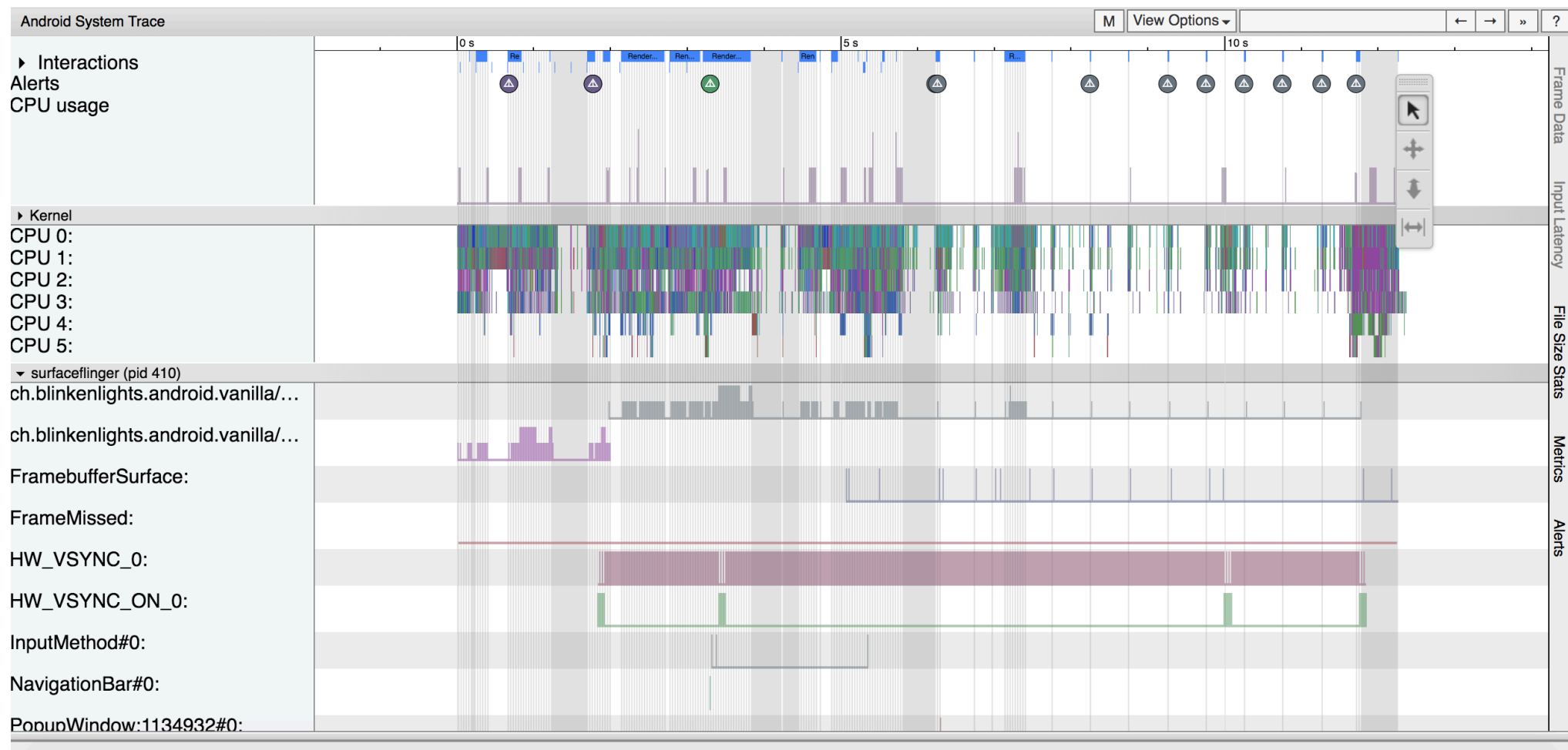
35.387825562 seconds time elapsed

**For more plots please see:**

**<https://github.com/ZiyeHan/CMPS-202/tree/master/throughput/results>**

# What Did I Run

## Latency



**python systrace.py --time=100 -o latencyResult.html**

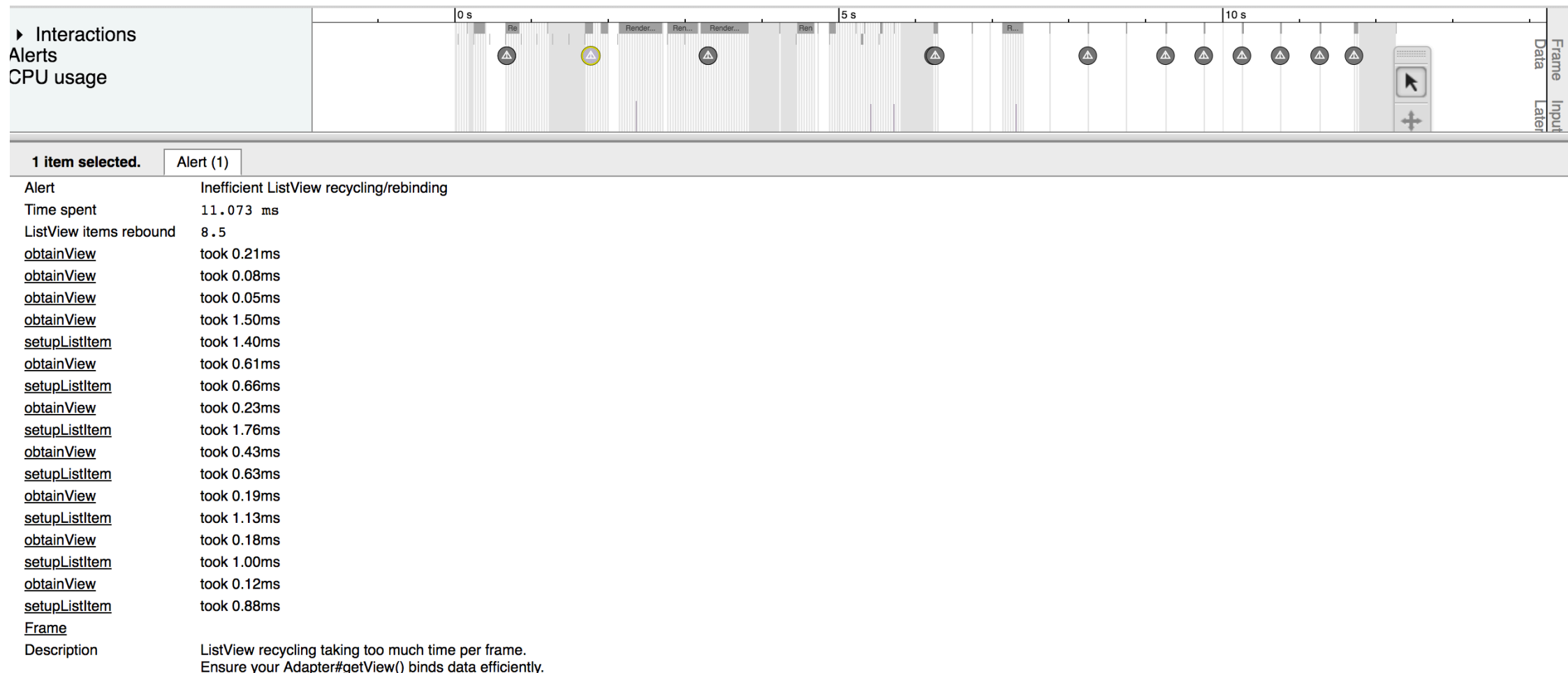
**Monitor 100s, then collect latency and generate report**

**adb shell monkey -p ch.blinkenlights.android.vanilla -v --throttle 200 500**

**Send 500 random actions to phone,**

# What Did I Run

## Latency



```
python systrace.py --time=100 -o latencyResult.html
```

**Monitor 100s, then collect latency and generate report**

```
adb shell monkey -p ch.blinkenlights.android.vanilla -v --throttle 200 500
```

**Send 500 random actions to phone,**

# What Did I Learn

- How to use docker
- How to use Perf
- How to use pmu-tools to run throughput test
- How to control program multi-thread
- Some linux commands
- How to use Monkey Runner to run automatic tests on App
- How to use systrace to generate HTML report of latency