

Are Mobile OS Ready for the Post-Smartphone Era?

林小竹 Felix Xiaozhu Lin

Assistant Professor, Purdue ECE

<http://xsel.rocks>

In Collaboration with Prof. **Feng Qian**, Indiana
University

Sponsors: NSF and Google

Post Smartphone Era

- Smartphones have changed the landscape of computing
- More diverse devices are coming
 - Wearables
 - IoT
 - AR/VR headsets
- System software is lagging behind

Top questions

- Wearables should enjoy
 - Baremetal performance
 - Baremetal efficiency
- Commodity smart watches
 - 80 million devices sold
 - How are they doing?

[MobiSys'16] “Understanding the Characteristics of Android Wear OS”,
Renju Liu and Felix Xiaozhu Lin.

A motivational case

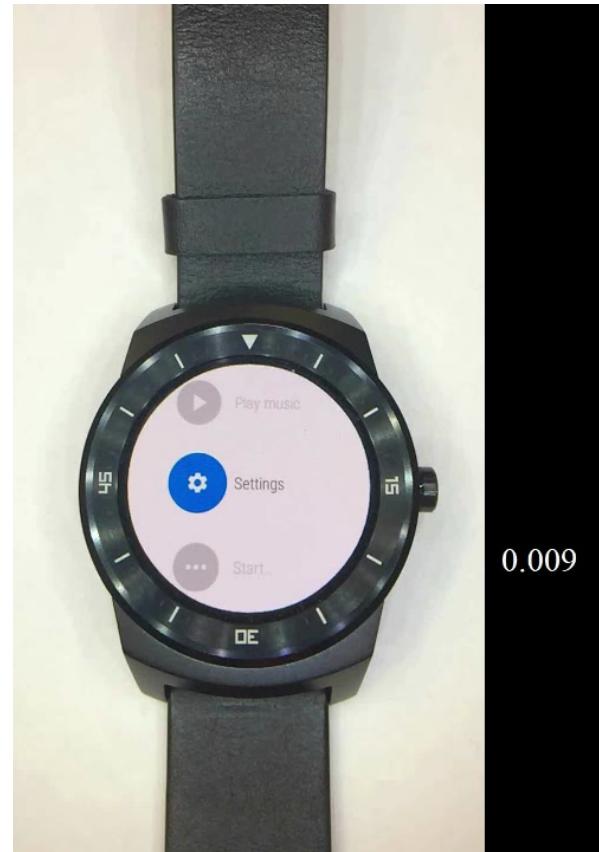
- The current performance & efficiency are far from baremetal
- **Pacing – inefficient**
 - face update: 400ms 88% busy

Clock face update

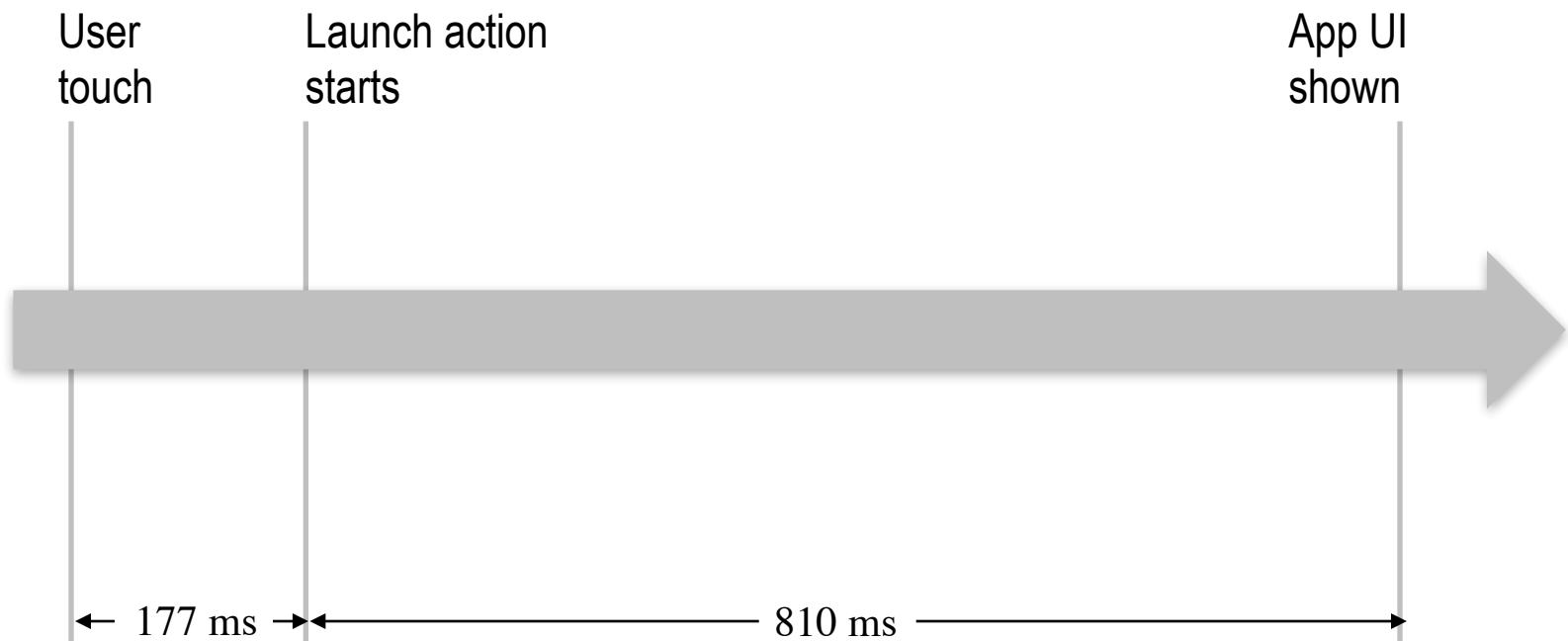


A motivational case

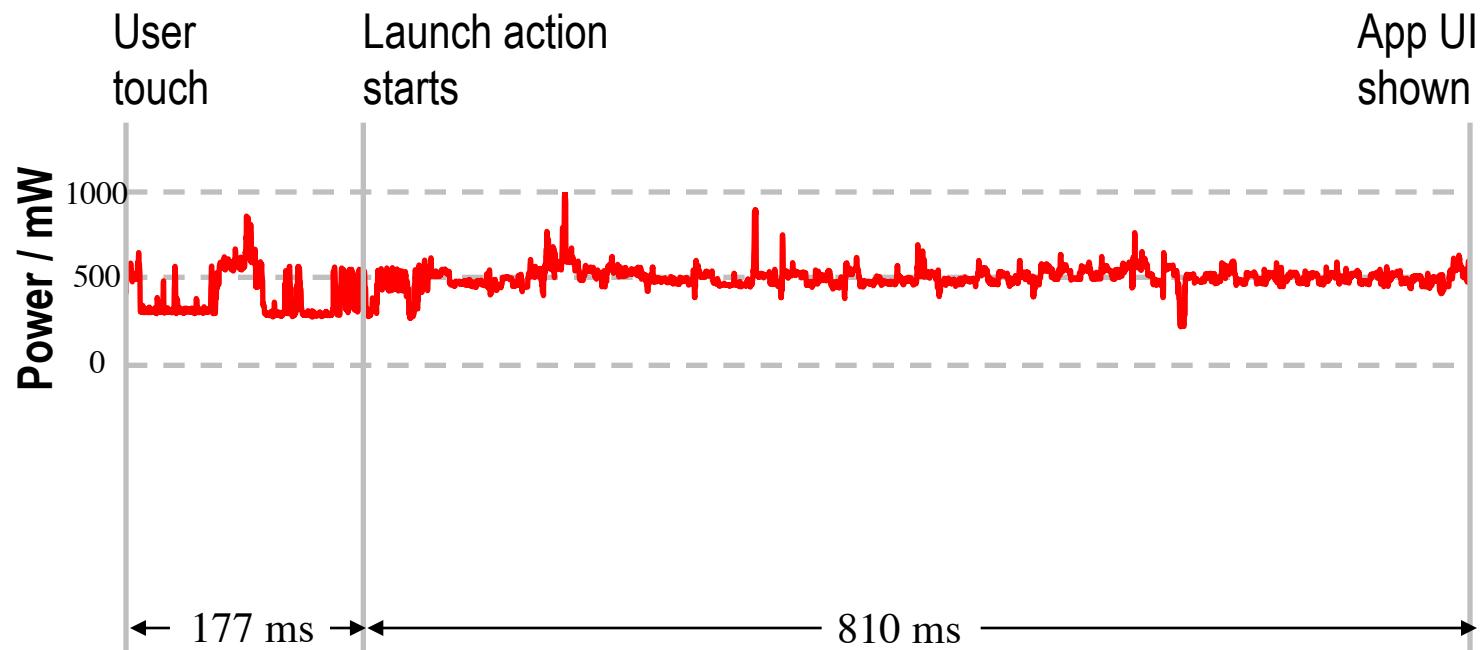
- The current performance & efficiency are far from baremetal Launch “settings”
- **Pacing – inefficient**
 - face update: 400ms 88% busy
- **Racing – slow**
 - Launch an in-mem app: 1 sec



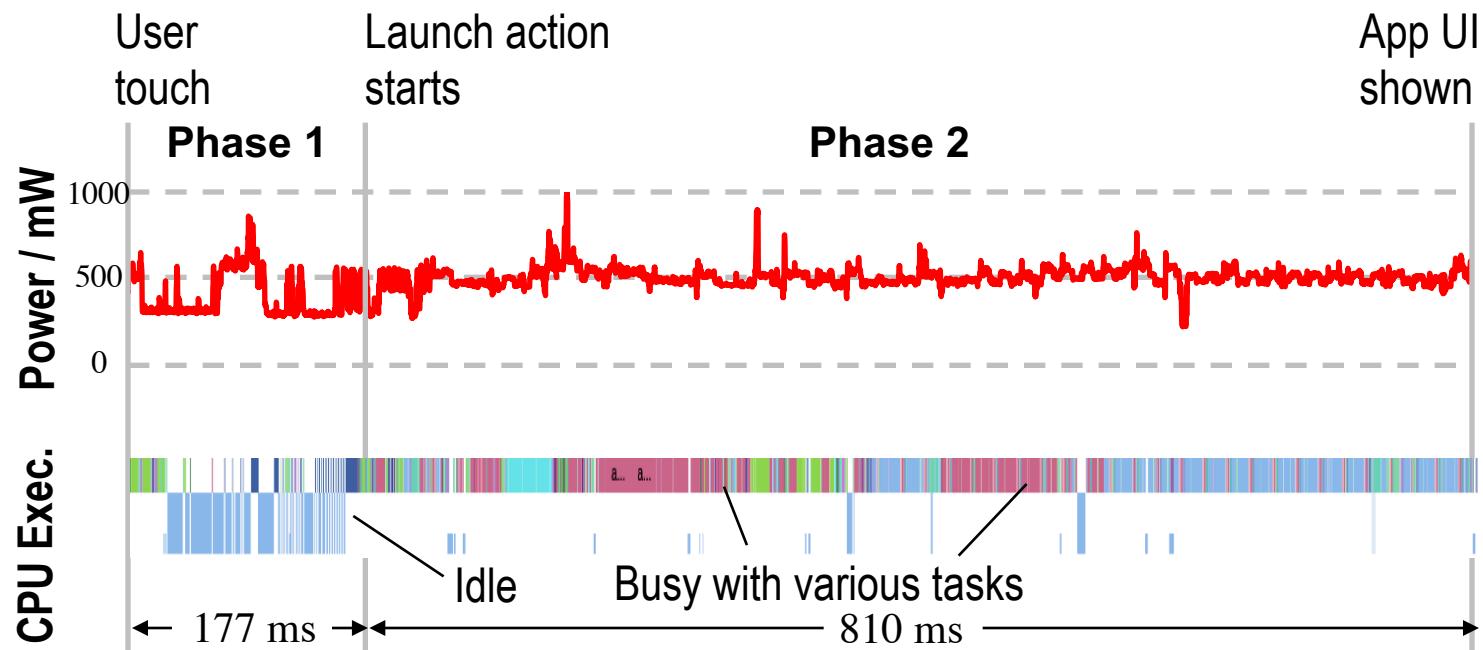
What happens underneath?



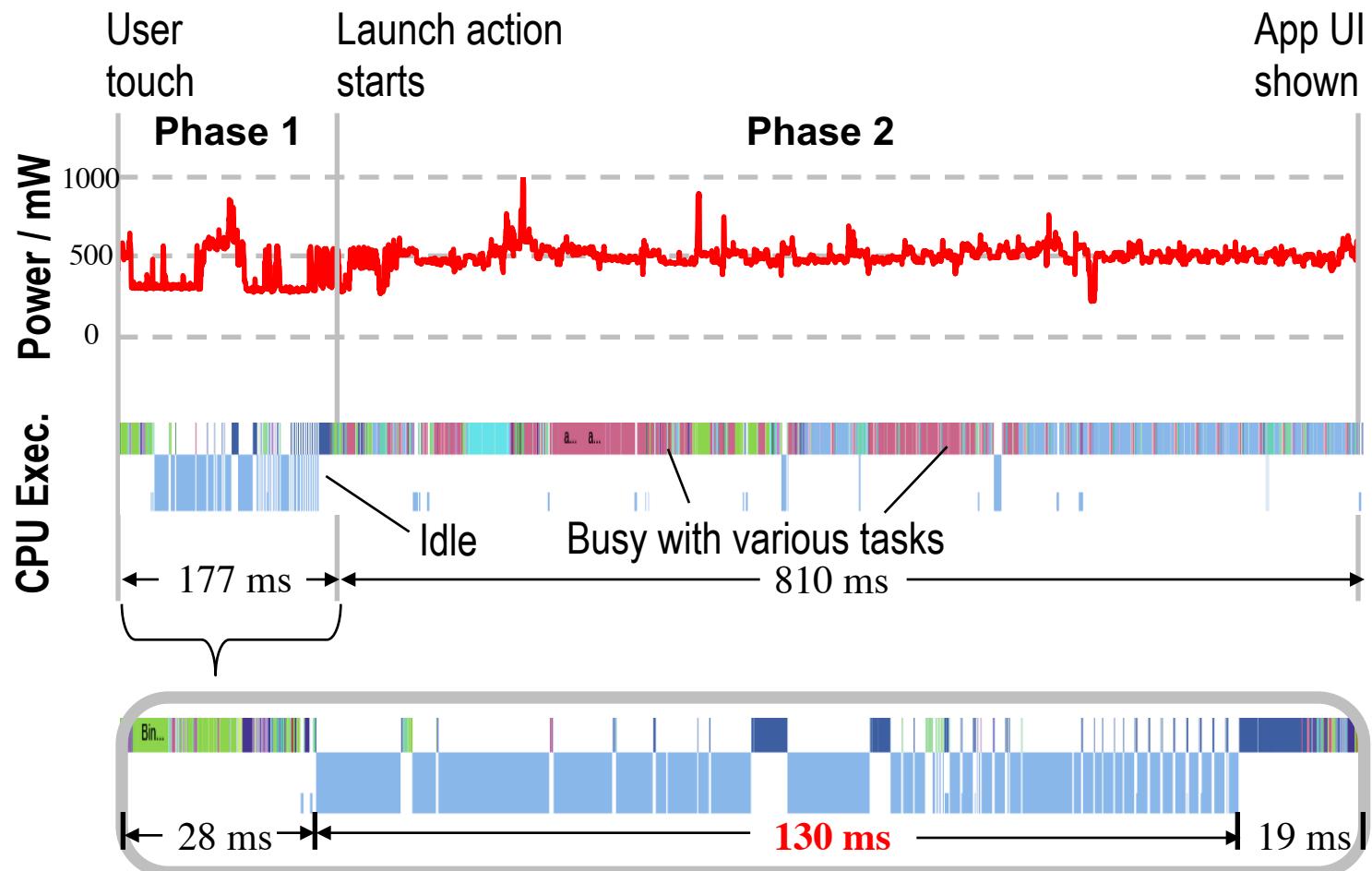
What happens underneath?



What happens underneath?



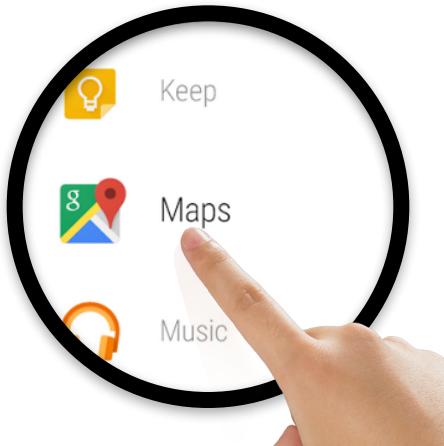
What happens underneath?



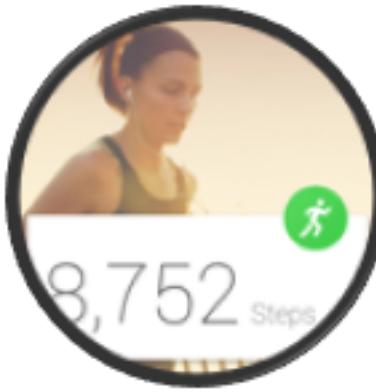
Profiling – Core use scenarios



Wakeup
Update
notification
wrist...



Single Input
launch apps
palming
voice...



Sensing
Accel
heart
baro



Interaction
Game
notes
navigation

Four Lessons

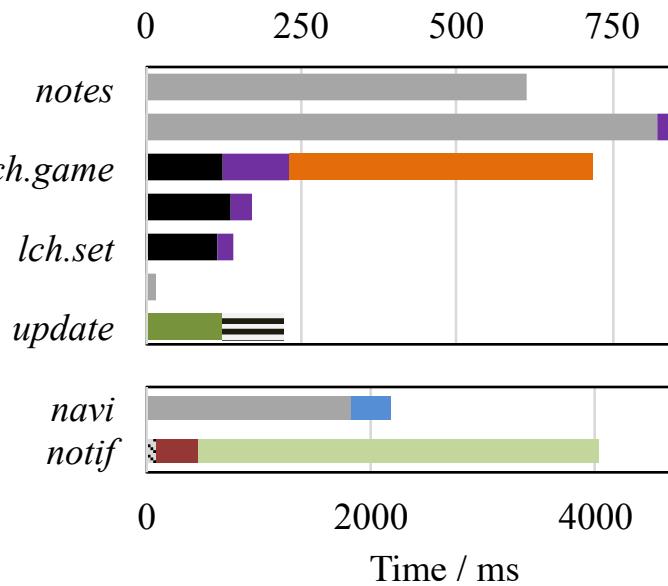
- Design for short interactions
- Clean the cruft
- Multicore is your friend
- Repair, don't rebuild (yet)

Four Lessons

- **Design for short interactions**
- Clean the cruft
- Multicore is your friend
- Repair, don't rebuild (yet)

Idle anomalies are pervasive

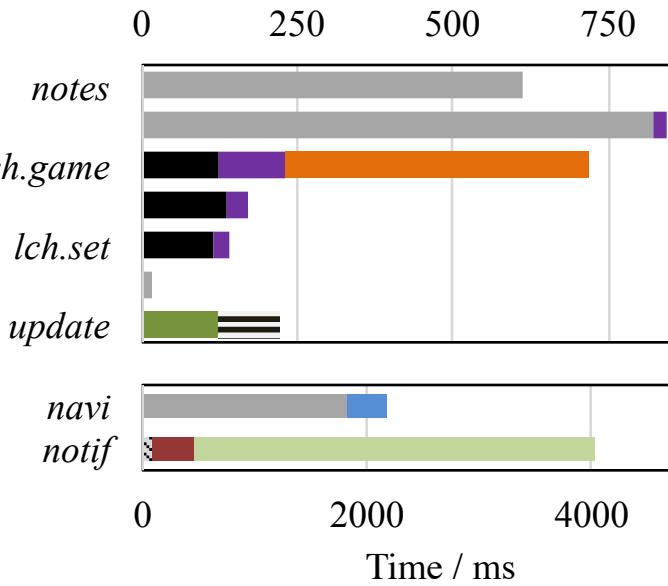
- Device suspend
- Voice UI
- Cont. interaction
- Cont. interact.+NetI/O
- Storage I/O
- User think
- Bluetooth tail time
- OS shell policy
- App policy



Time (ms)	Pct. Overall	Episodes	Pct. Explained
614.1	17.1%	376	100.0%
843.3	50.5%	352	100.0%
722.6	50.9%	205	99.9%
185.2	25.6%	110	92.9%
153.6	15.6%	120	91.4%
16.8	10.6%	6	100.0%
223.0	61.2%	44	100.0%
2173.0	52.80%	912	100.0%
4035.6	86.80%	277	100.0%

Idle anomalies are pervasive

- Device suspend
- Voice UI
- Cont. interaction
- Cont. interact.+NetI/O
- Storage I/O
- User think
- Bluetooth tail time
- OS shell policy
- App policy



Time (ms)	Pct. Overall	Episodes	Pct. Explained
614.1	17.1%	376	100.0%
843.3	50.5%	352	100.0%
722.6	50.9%	205	99.9%
185.2	25.6%	110	92.9%
153.6	15.6%	120	91.4%
16.8	10.6%	6	100.0%
223.0	61.2%	44	100.0%
2173.0	52.80%	912	100.0%
4035.6	86.80%	277	100.0%

Legacy/improper OS designs

Anecdote

Voice UI

Four Lessons

- Design for short interactions
- **Clean the cruft**
- Multicore is your friend
- Repair, don't rebuild (yet)

Four Lessons

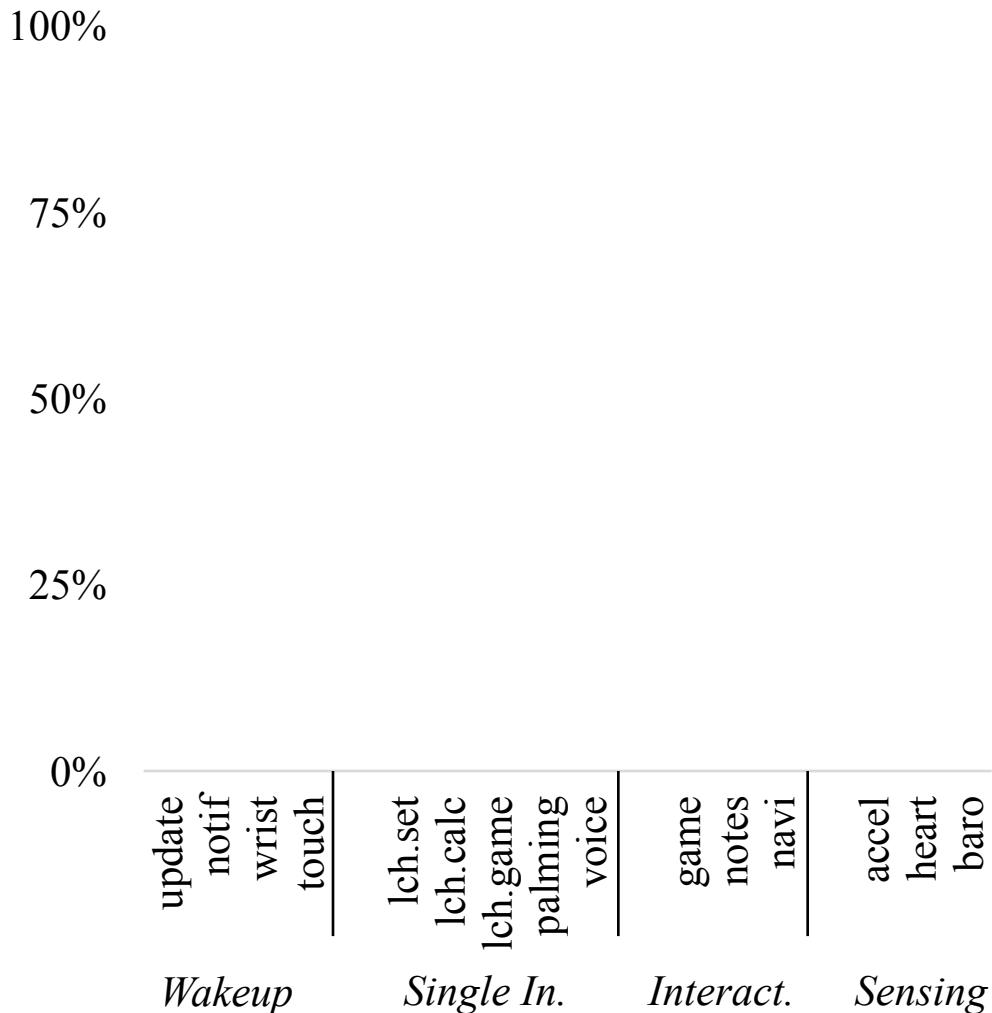
- Design for short interactions
- **Clean the cruft**

cruft  (krūft)

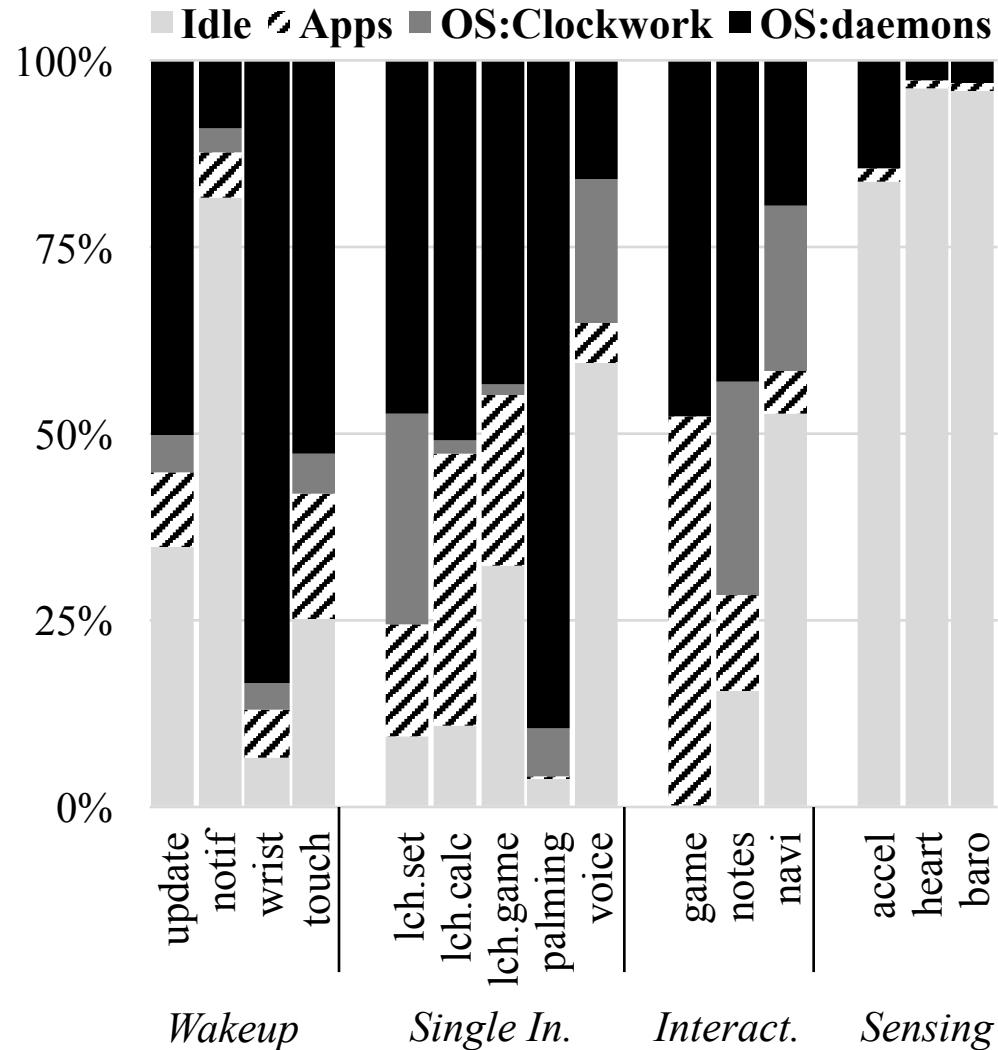
n.

1. Trash, debris, or other unwanted matter that accumulates over time.
2. Unnecessary digital information that accumulates over time, such as unneeded files or obsolete lines of code in software: *"By removing cruft, you can recover valuable disk space ... and reduce the chance of software conflicts" (Joe Kissell).*

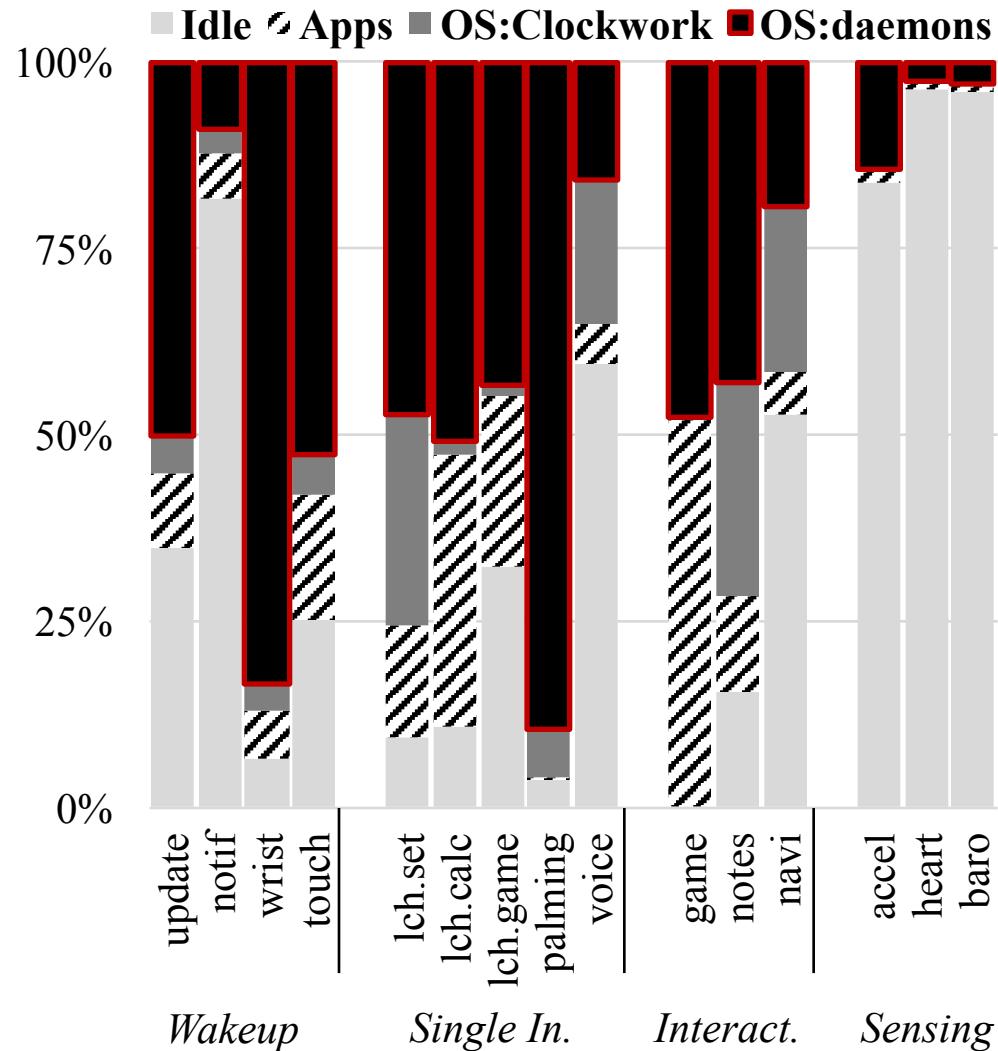
OS execution dominates CPU usage.



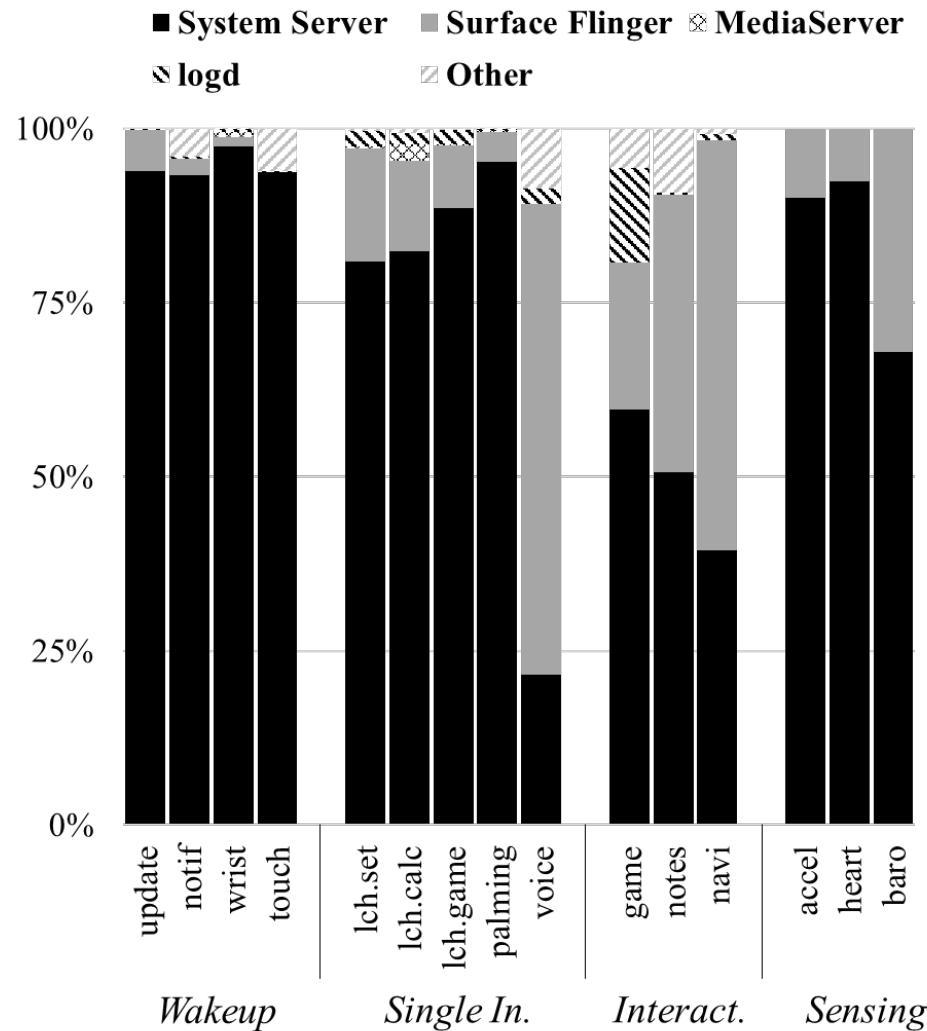
OS execution dominates CPU usage.



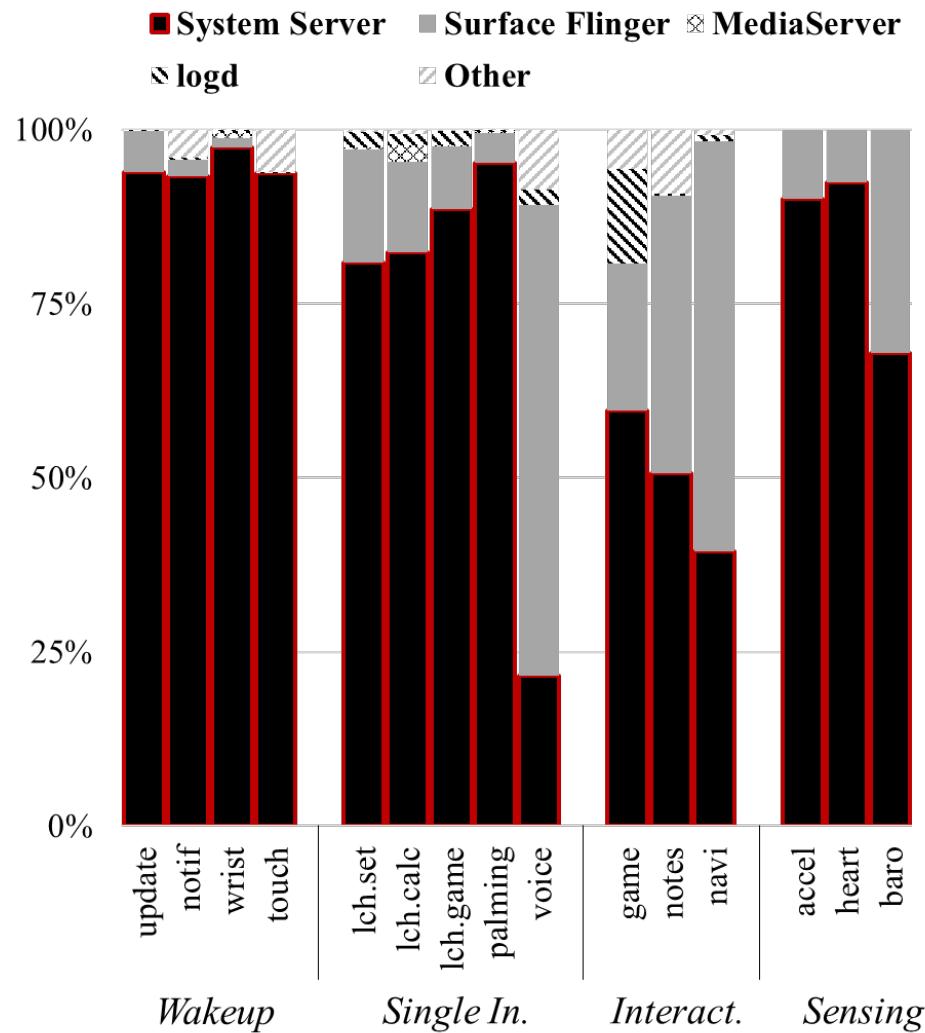
OS execution dominates CPU usage.



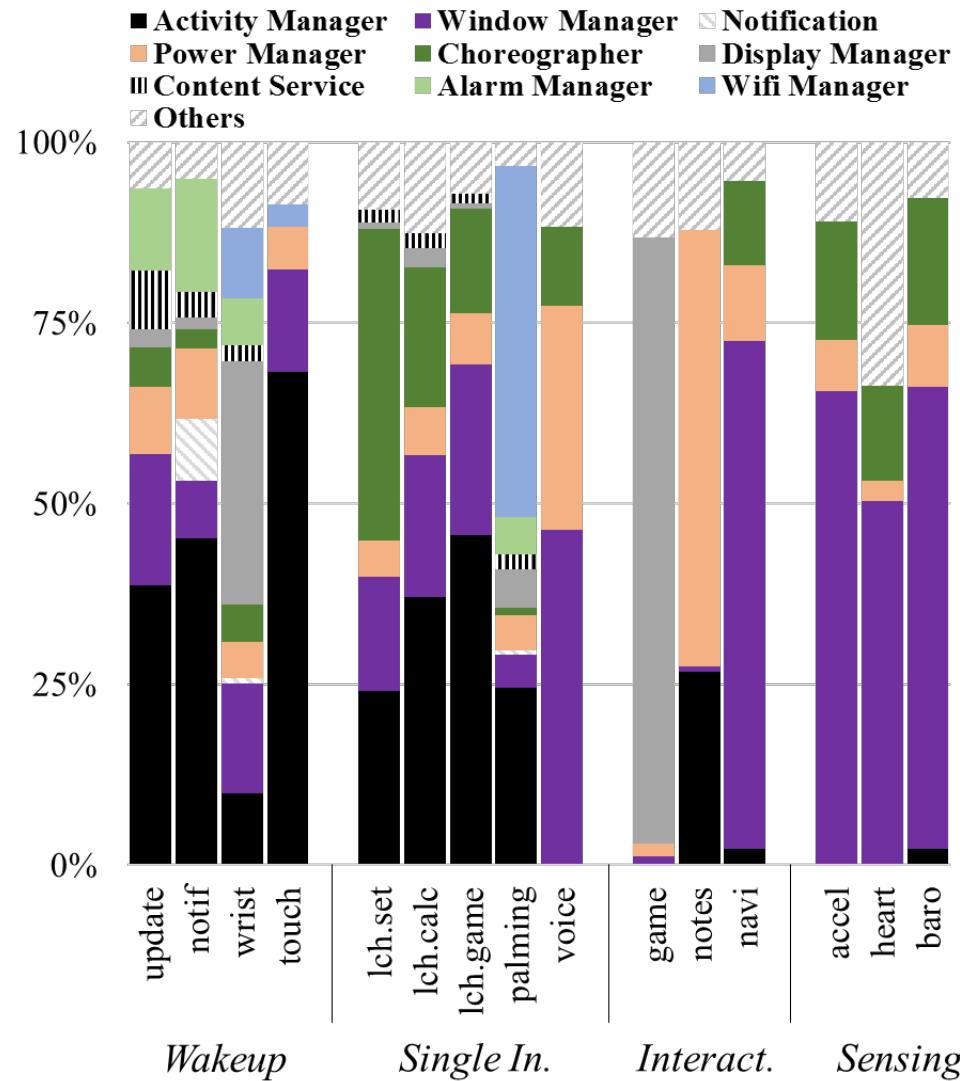
OS execution dominates CPU usage.



OS execution dominates CPU usage.

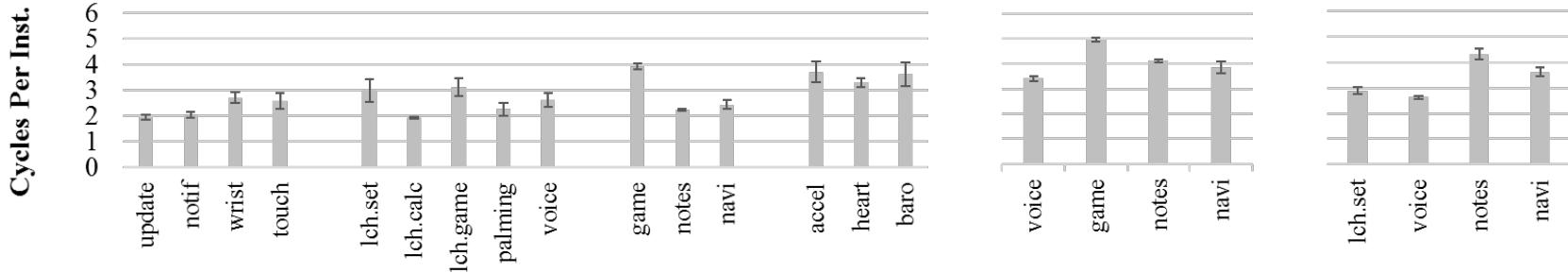


Costly OS services are likely cruft.



Wearable suffers from uArch inefficiency

Cycles-per-instruction (lower is better)
2 -- 5 (high!)



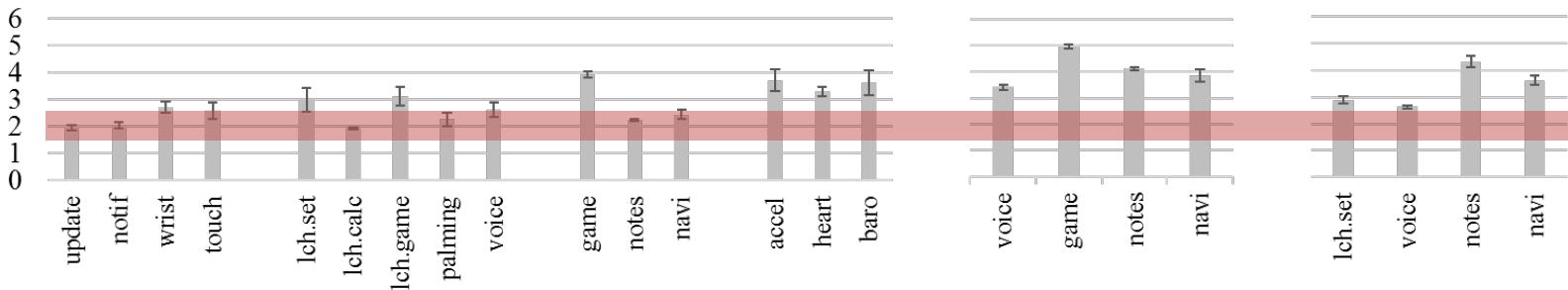
Wearable suffers from uArch inefficiency

Cycles-per-instruction (lower is better)
2 -- 5 (high!)

Smartphone as a comparison

1.3 -- 2.5 web rendering

Cycles Per Inst.



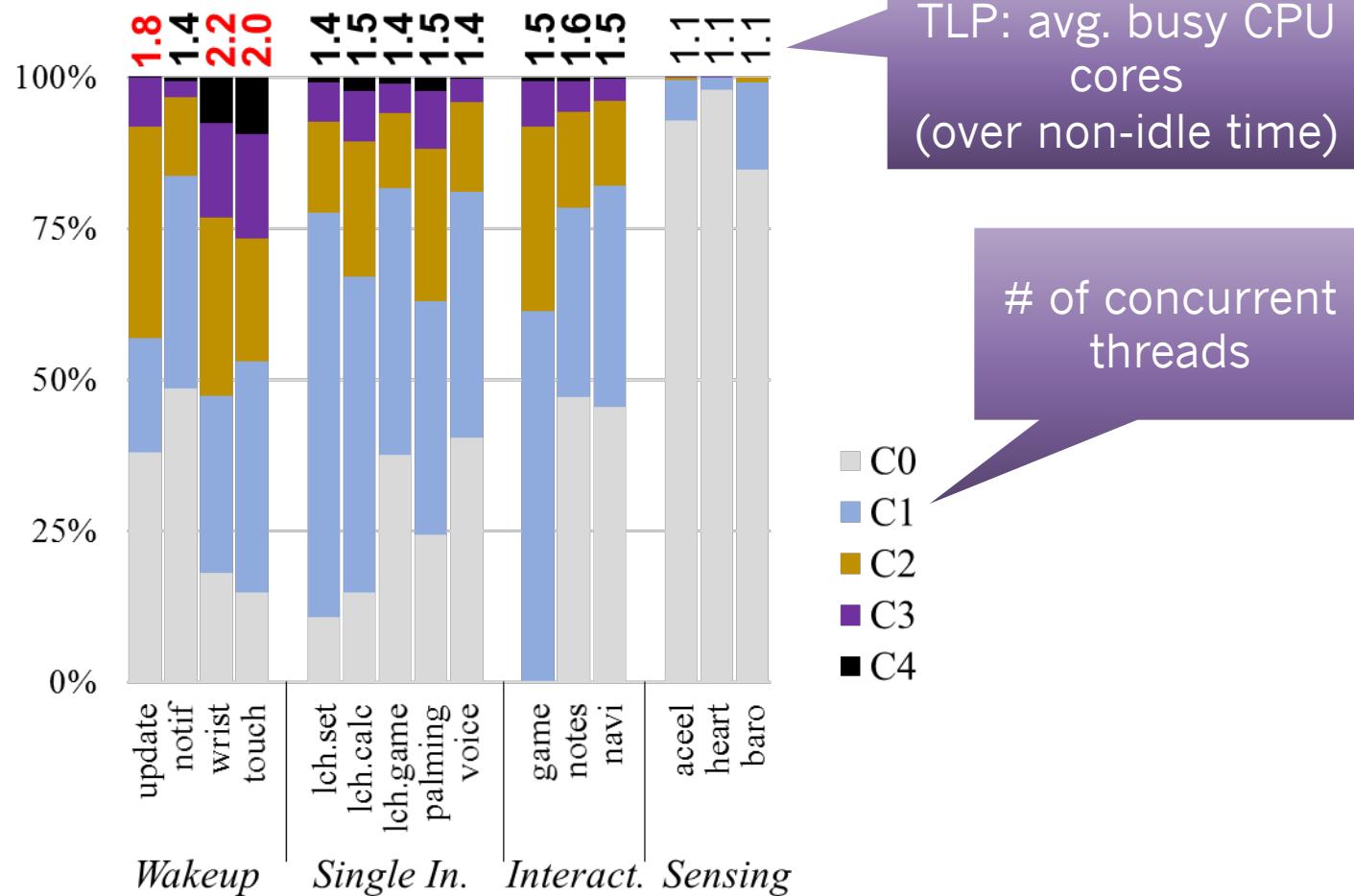
**The major cause: complex OS code
(L1 icache, iTLB, and branch predictor)**

**uArch problem will NOT be gone with
future wearable CPUs**

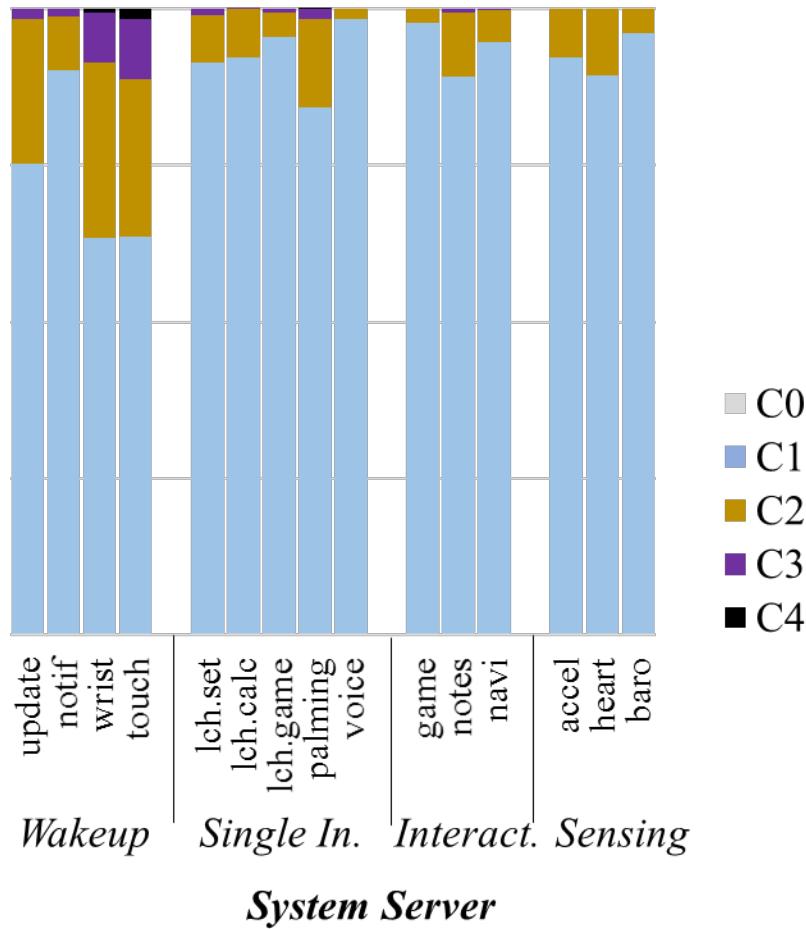
Four Lessons

- Design for short interactions
- Clean the cruft
- **Multicore is your friend**
- Repair, don't rebuild (yet)

Substantial TLP on a par with desktop -- due to short interactions.



Apps are mostly single-threaded; OS contributes to TLP significantly.



Four Lessons

- Design for short interactions
- Clean the cruft
- Multicore is your friend
- **Repair, don't rebuild (yet)**

Hot OS daemon functions: highly skewed distribution

Top 5 → >20% CPU cycles

Top 50 → >50% CPU cycles

Hot OS daemon functions: highly skewed distribution

Top 5 → >20% CPU cycles

Top 50 → >50% CPU cycles

Manipulating basic data structures

Legacy/improper OS designs

Anecdotes

Backlight UI layout low-mem killer

Summary: four lessons

- **Design for short interactions**
 - User's attention is precious
- **Clean the cruft**
 - which wastes CPU cycles and chokes uArch
- **Multicore is your friend**
 - OS is already multi-threaded; keep it that way
- **Repair, don't rebuild (yet)**
 - Java is fine; Dalvik is fine; ARMv7a is fine...
 - Fix individual OS components first

**HOW ABOUT AFTER THAT?
(I.E. “NEXT-GEN WEARABLE OS”)**

Computational Jewelry: Amulet

Dartmouth College and Clemson University

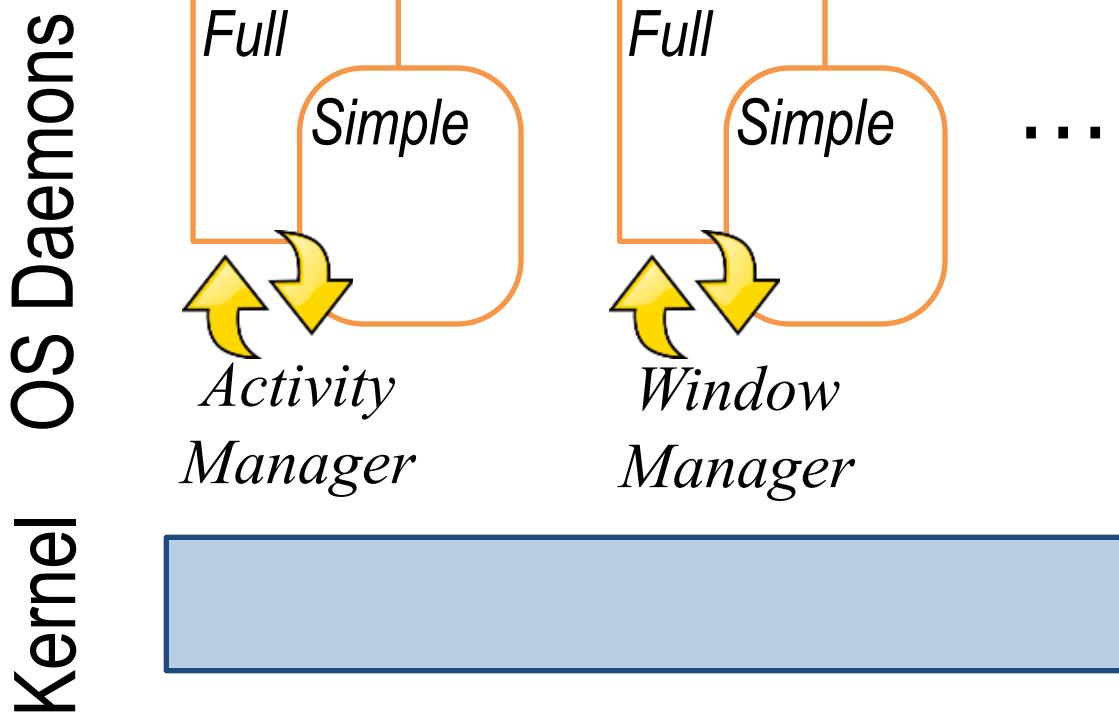


We probably will reach a point when OS overhaul/redesign is justified.

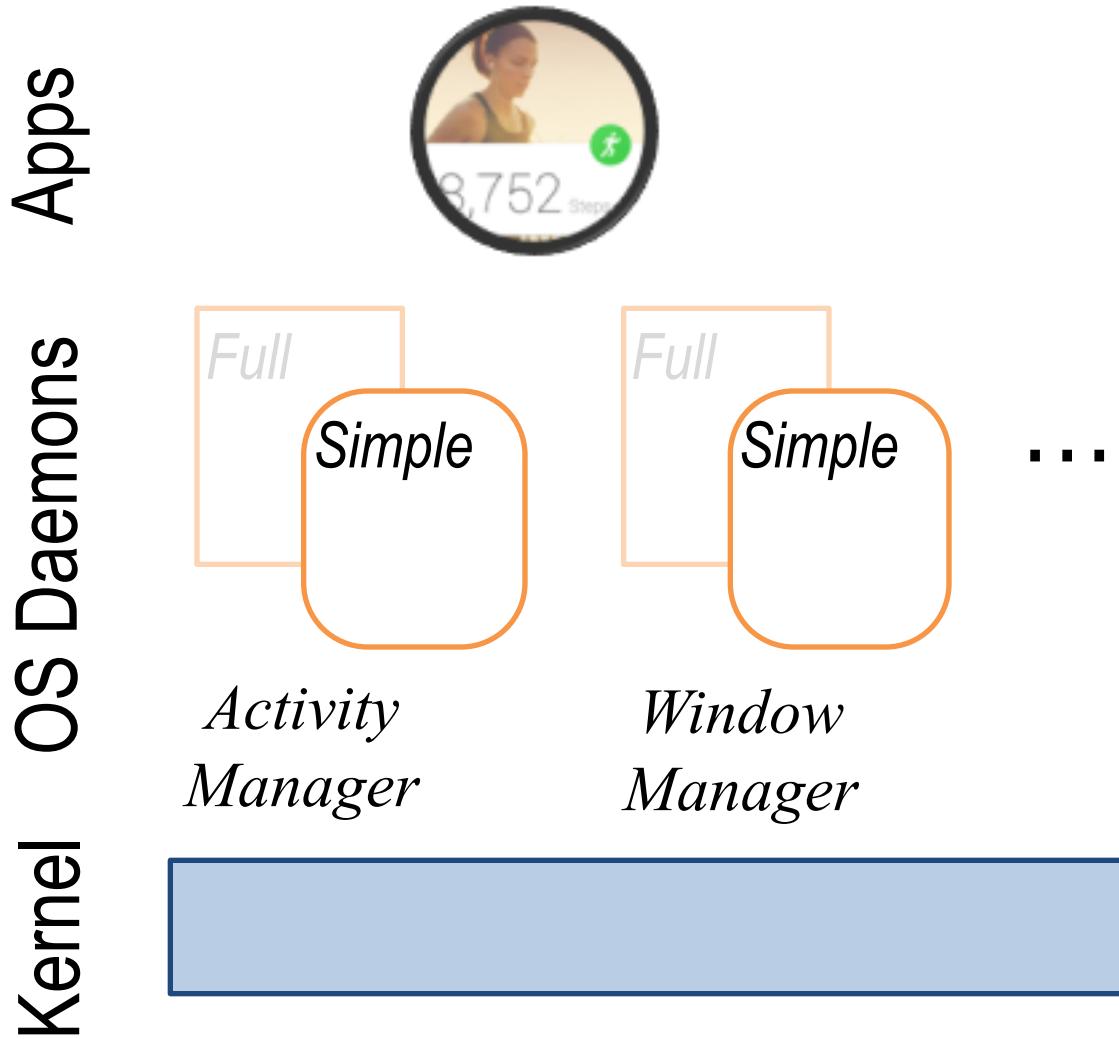


Specializing OS
for common, single-app **scenarios**

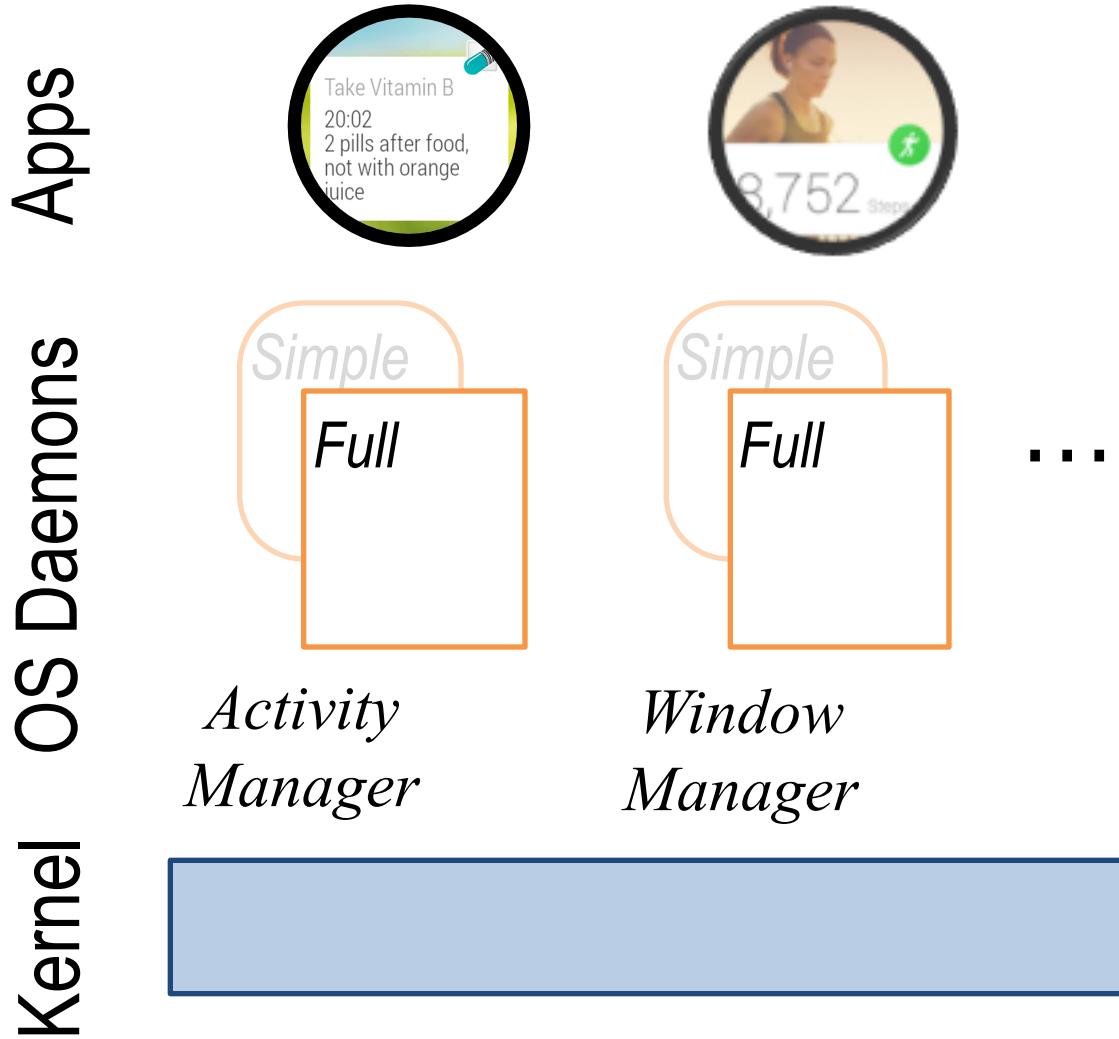
“In-place” OS specialization



“In-place” OS specialization



“In-place” OS specialization



Final takeaway

- Post-smartphone devices: unique usage and hardware
- Many OS tradeoffs become invalid
 - efficiency v.s. flexibility & programming ease
- Immediate actions: reusing/fixing individual components
- Future: specialization



Tools, data, and benchmark videos

xsel.rocks/p/wear

Purdue ECE

- Largest engineering dept in Purdue
- 127 years
- Consistently ranked as one US top 10
- ~90 faculty members; ~500 PhD students
- 28 IEEE fellows, 4 NAE members and more
- \$31M annual research fund

Purdue ECE Is Unique

- Diverse & Strong Computer Engineering
 - ~20 faculty members
 - Hardcore computer research (OS, PL, compiler, architecture, mobile/emb system)
- Active in various communities
 - ISCA/MICRO/HPCA, PLDI/PPoPP, ASPLOS, Mobicom/Mobisys, etc...