Motivation: Problem of the Pull Model

- Power consumption in an idle, mobile device can be attributed largely to an event model known as the "pull" model, which repeatedly queries the operating system and hardware to determine if events exist.

Research Approaches

- Take advantage of new mobile device hardware that allows
  - Devices to "push" events to OS when events occur
  - CPU to enter "deep sleep" states that conserve power
  - Tasks to be distributed to multiple cores

Use the Android OS and NVIDIA TK1 for benchmarking and prototyping

- Pull models implemented in the Android operating system all utilize a polling loop to continuously check for the existence of an event[1].

CURRENT PROGRESS

- ESM has been implemented for the keyboard and mouse event portions of Android's OS.
  - Versus a pull model with a 16 millisecond delay (current state of the art for Android OS), ESM reduces power consumption by
    - ~250mW for infrequent events
    - ~120mW for frequent events.
  - Versus a pull model with a 100 millisecond poll loop, ESM reduces power consumption by
    - ~75mW for infrequent events
    - ~50mW for frequent events.

ABSTRACT

- A mobile device must be recharged on a daily basis even if it was in the user’s pocket all day[1].
- Many apps needlessly drain the battery when you are not interacting with your device and prevent the CPU from being placed in energy-conserving "sleep" states[1].
- Problem Source: Mobile OS’s use an event handling "pull" model, where an application must constantly poll to see if events are present, thus keeping the CPU aroused[1].
- Problem Solution: We are developing a "push" model, where the CPU sleeps until events occur—then the OS awakens the appropriate app and executes it.
- Preliminary results with the GUI portion of a mobile app shows large power savings and a reduction in latency (the time between an event’s occurrence and an application’s response).

FUTURE DIRECTIONS

- Use the ESM to improve power consumption in other power hungry areas, such as 4G or WiFi.
- Use the ESM to improve power consumption in adjacent areas, such as physical network layers.
- Move display server from application layer to OS so that knowledge provided by ESM can be used to facilitate scheduling of apps.
- OS might shut down WiFi if foreground app does not require it
- Drawing threads for background apps might be suspended
- Distribute event handling across cores: Use a lower-power consuming CPU core for OS event handling and higher power cores for application event handling.
- Examine middleware layers, such as .NET and Java to streamline their event propagation models.
- Measure the benefits of the ESM on cutting edge, 64-bit mobile CPUs, such as the NVIDIA TX1 system.

CONCLUSIONS

- Prototyping using the Android OS shows empirically that our model significantly reduces power consumption and latency when the device is somewhat or mostly idle.
- Power consumption and latency is still reduced when the device is receiving many events rapidly.
- Our ESM has room to grow and needs further research to apply to other event-oriented systems that might needlessly consume power, such as WiFi or 4G (cellular).

REFERENCES