Michael Tang
Due 02/20/2019 Paper Summaries: Potpourri

E06 - Smartphone Energy Drain in the Wild Chen et al. seek to create a detailed breakdown of smartphone energy uses between apps, OS and base hardware operations. This way, developers of each of these fields can know their onus in extending battery life for their users. They used separate metrics for potential power usage to calculate draw - frequency & utilization for CPU/GPU, brightness for screen and FSM & signal strength for networking. These draw calculations are also partitioned for various app and OS activities. Specifically, power draw measurements are logged every 5 minutes for app-wise CPU usage, every 1-5 seconds for background usage and on demand for hardware switching and logging overhead. Observations were then modeled to distributions to create daily average energy use breakdown. Chen et al. concluded that the biggest usage (and therefore biggest burden) is related to screen as well as networking apparatus. F03 - Stanley: The Robot that Won the DARPA Grand Challenge The team from Stanford, Volkswagen, Intel, and Mohr Davidson behind Stanley detail the components that went behind Stanley's winning race. Though good engineering was still needed to perfect drive-by-wire and the mounted technology on the car, the researchers determined early on that the vision software would be most vital to develop. Terrain data was pipelined from the sensor interface of lasers, radar, and GPS reading the terrain and position directly, through the perception layer of mapping software to direct control of the car and user input. External servers also played a vital role in data logging and time metrics. After receiving sensor data, the onboard computer creates an estimation of the car's current locational state via GPS, and probabilistically during GPS outages. This combined with terrain mapping allows the control layer to predict future location and make velocity and steering adjustments accordingly. G03 - Optimizing 360 Video Delivery over cellular networks Qian et al. establish that with the advent of mobile VR and 360 videos accessible to mobile devices, the ability of cellular networks to load these videos will become important. For a comfortable viewing experience, videos must be streamed at 1080s or higher consistently. This is a problem that would be alleviated by increasing network throughput and reliability, but Qian et al. want to maximize optimization by also changing delivery methodology. The most immediate and promising solution is spacial buffering on top of the temporal buffering that web video services conduct. This way, the entire 360 degree space does not have to be loaded at each second. To aid that, Qian et al. look into predicting head movement of the user in a timeframe of 1-2s, more than sufficient time for today's LTE networks to stream those upcoming frames. Using weighted linear regression they achieve an encouraging accuracy of 92.4% with a margin of error of 3.7% for head movement in the next second. Qian et al. conclude with future prospects, including a large-scale Android streaming app they are developing based on their findings in this paper, based on HTTP and able to integrate smoothly with present video services. V02 - Real-time Video Analytics Ananthanarayanan et al. claim that pooled computing in a geographically distributed architecture of public clouds, private clusters, and edge devices themselves must be implemented for the most optimal real-time video analytics. This is due to the standards of low latency required for real-time as well as the high quality video of the future requiring higher bandwidth usage. Thus network usage must be both alleviated via prior filtering and computations at the edge device as well as distributed queries, which would perfectly fit the edge computing structure described above. To that end they propose Rocket, their video analytics stack, which emphasizes onboard activities to reduce computations such as electronic camera steering, ability for multiple concurrent queries on cameras, and smart feed selection. Network infrastructure, they establish, must be geographically well distributed, able to handle many queries, and built heterogeneously to handle different capacities of workload. Ananthanarayanan et al. envision that Rocket and future iterations can be applied to countless important applications such as self-driving, security, and alternate reality.
Summary This section's theme was a little about the cool ways mobile computing can be applied to the world at hand, but was much more about good methodology when designing mobile computing systems. All of these researchers were trying to optimize their systems built to accomplish a task, and they first identified the key ways to obtain that optimization. Focusing on those ways led to great results for all of them, and promising immediate future improvement for all of us.