Scribe Notes for Lecture 20 - November 6th

Power Management of Online Data Intensive Services
David Meisner, Christopher M. Sadler, Luiz André Barroso, Wolf-Dietrich Weber and Thomas F. Wenisch

1. **Online Data-Intensive (OLDI) Services**
   1.1. Large-scale workloads (e.g. online advertising, machine learning, etc)
   1.2. Workloads are user-driven via queries
   1.3. Responsiveness (end-to-end latency) is key performance metric
   1.4. Large utilization variation of workload throughout the day

2. **Energy Proportionality**
   2.1. Power consumption is directly proportional to load
   2.2. Achieved through power management mechanisms that trade off power and performance (i.e. P- and C-states, etc.)
   2.3. Mobile devices demonstrate high energy proportionality
   2.4. OLDI services are not energy proportional
   2.5. Became mainstream in: http://dl.acm.org/citation.cfm?id=1339894

3. **OLDI Characterization**
   3.1. Focused on energy proportionality from processor, memory, disc and their interactions with one another
   3.2. Production google Web-search workload
   3.3. Results are presented as activity graphs which show CDFs for percentage of time a system is idle across different utilization levels
   3.4. Identify effectiveness of current power management mechanisms for each of the components

4. **Overview of Power Management Mechanisms**
   4.1. Overview of Intel C/G/S/P states can be found here: http://www.overclock.net/t/1058894/intel-acpi-guide-c-g-s-p-states-and-ocs
   4.2. Power Down Modes
      4.2.1. Active mode: peak component operation
      4.2.2. Doze mode: component operation halted, some subsystems are powered down
      4.2.3. Nap mode: component operation halted, more subsystems are powered down
      4.2.4. Sleep mode: component operation halted, component completely shut down
   4.3. Gating Mechanisms
      4.3.1. Power gating: component is powered down
      4.3.2. Clock gating: halt clocking component

        4.3.3. Data gating: unused data is not toggled
    4.4. Dynamic Voltage and Frequency Scaling
        4.4.1. Adapt voltage and frequency at runtime
        4.4.2. Software-controlled governors in Linux: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt
        4.4.3. Hardware-controlled mechanisms also exist (i.e. Intel TurboBoost)

5. Power and Performance Model
- 5.1. Single server system is studied in a test harness to emulate datacenter queries
- 5.2. Created (and validated) performance model for the effects of different power management techniques on query responsiveness
- 5.3. Leaf Node Performance Model
    - 5.3.1. L_query = L_service + L_wait
        - 5.3.1.1. L_service is time to service request
        - 5.3.1.2. L_wait is random processes for when query can be serviced

6. Paper Conclusions
- 6.1. The best component to optimize in isolation is the CPU for energy proportionality
- 6.2. CPU core-level power management mechanisms are good
- 6.3. Need to extend current CPU power management mechanisms to memory hierarchy (caches and memory controller)
- 6.4. Memory is underutilized and thus shows promise as an effective place to implement power management techniques
- 6.5. Power down techniques (i.e. PowerNap) are not effective for OLDI workloads
- 6.6. Must deliver tail latency in energy-proportional manner
- 6.7. Full-system power management collaboration is also promising

7. Queuing Theory 101 will be presented next class