Use Energy Less and Wisely

Kang G. Shin

Department of EECS

The University of Michigan

HP Sustainable IT Workshop—10/20/08

Key Questions and Approaches

- How to use only the amount of energy required by the underlying applications without any waste?

  =>Top-down approach

- How to extend the lifetime of energy source?=> Manage it well

Use only as much as you need

Top-down approach

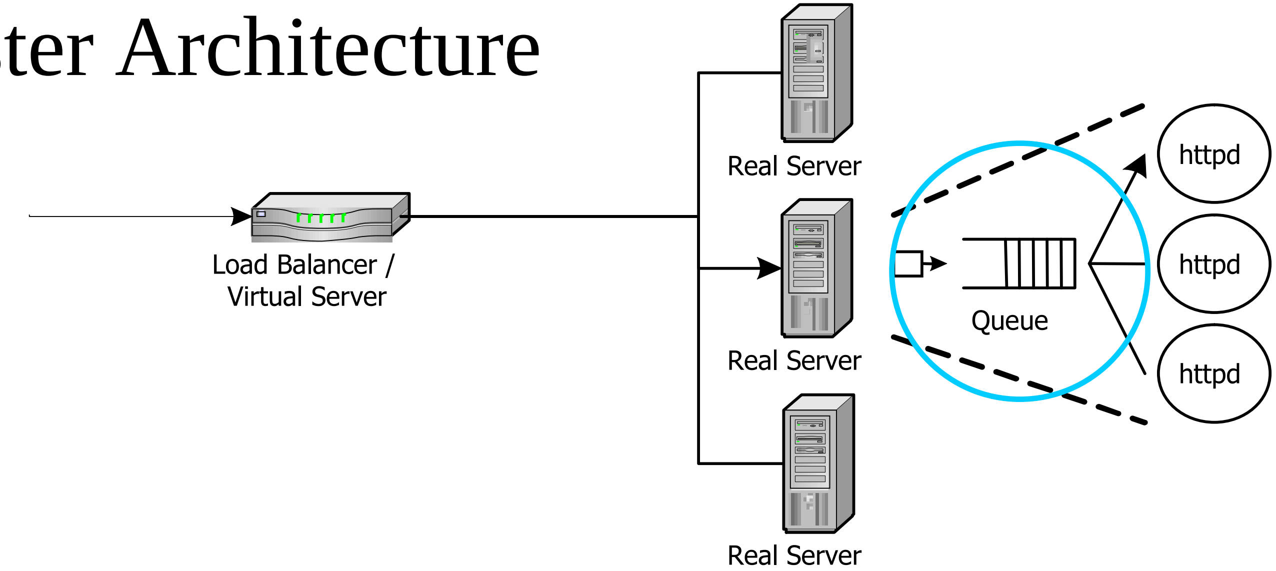
- Server clusters
- Single Server
- Server components
- ….

Server Clusters

- Server clusters must provide availability and capacity needed by applications.
- Capacity is usually statically and conservatively estimated to handle peak load, but...
  - Servers are under-utilized (excessive equipment purchase)
  - Powering and cooling costs are increasing (recurring cost)
  - Capacity need changes dynamically
- With accurate estimation of capacity need, datacenters can dynamically allocate resources to services/apps.
  - Improve utilization (can host more services/apps)
  - Reduce energy cost (turn off unused servers)
  - Ensure service quality (meet SLAs)

Server Cluster Model

- Cluster Architecture



- Sustainable throughput := maximal throughput subject to meeting certain thresholds of performance metrics
- Capacity is determined by
  - Server configuration (hardware, software)
  - Request arrival process, type of requests (dynamic, static)
  - Users' performance expectations (patience)
  - Queue size

Definition of Capacity

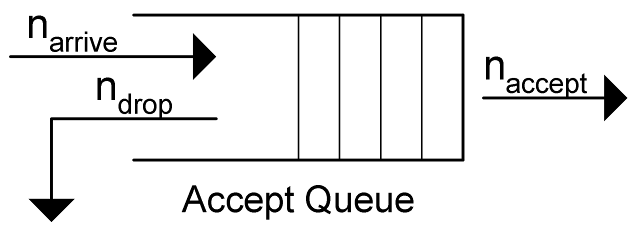
Capacity

The max throughput while keeping the request drop ratio under a customer-specified threshold

Request Drop Ratio = # of failed connection attempts / # of connection attempts

Queue Monitoring for Capacity Estimation

- Avoiding SYN retransmissions is key to dynamic cluster sizing
  - Need to predict when retransmissions become likely, i.e., when capacity is too low



- Estimate time needed to fill the queue

$$t_{fill} = \begin{cases} K / (\lambda_{\max} - c) & \text{if } \lambda_{\max} > c \\ \text{undefined} & \text{otherwise} \end{cases}$$

- Estimate avg queue length that causes overflows

$$\ell_{th} = \begin{cases} \dfrac{K \cdot \lambda_{\max} \cdot t_{fill}}{2(c(T - t_{fill}) + \lambda_{\max} \cdot t_{fill})} & \text{if } t_{fill} \le T \\ \dfrac{[K - (\lambda_{\max} - c)T] + K}{2} & \text{if } t_{fill} > T \end{cases}$$

- Capacity is estimated by

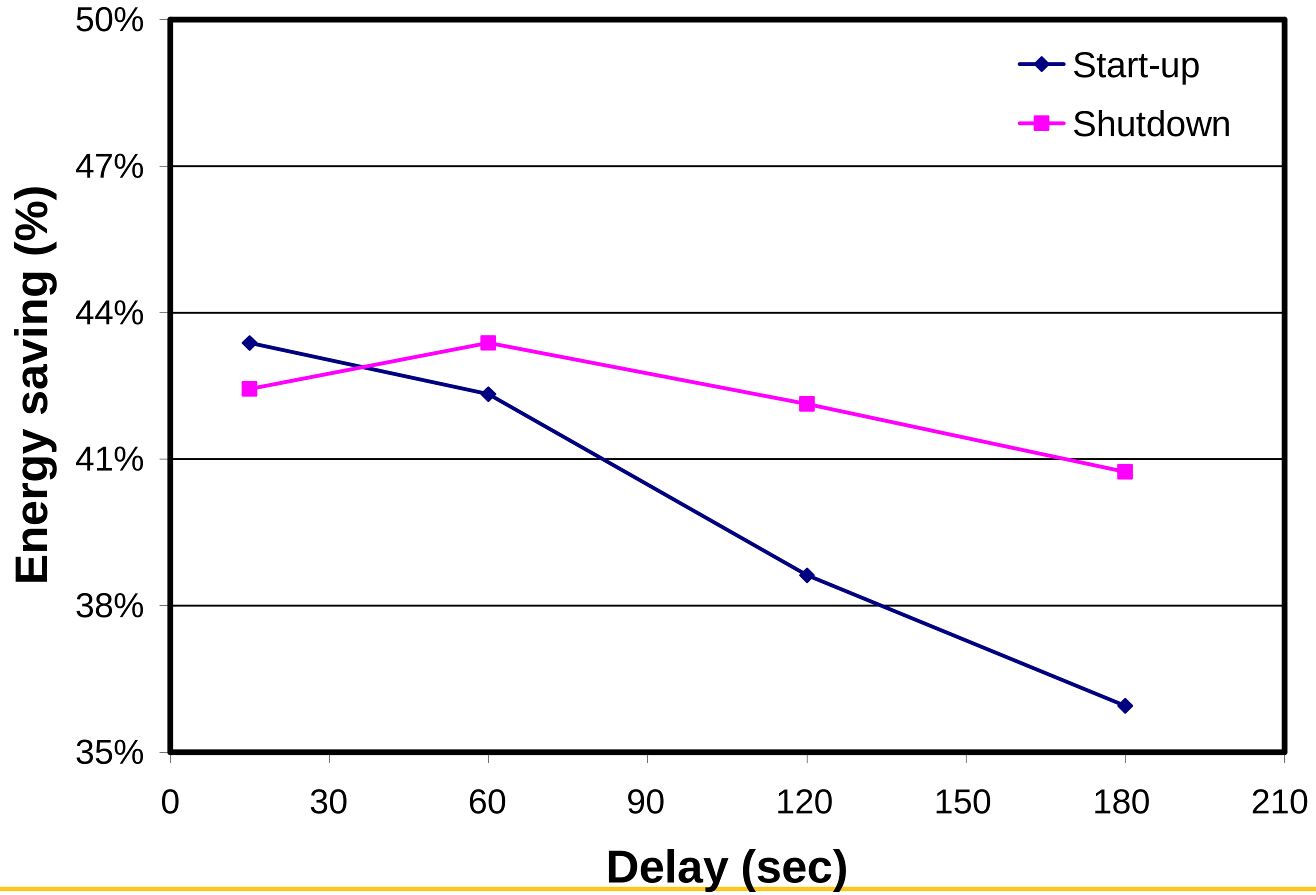
$$c' = \begin{cases} \mu & \text{if } \varepsilon \ge \varepsilon_{th} \\ c \cdot \alpha + \mu \cdot (1 - \alpha) & \text{if } \varepsilon < \varepsilon_{th} \text{ and } \begin{cases} \min(\bar{\ell}, \hat{\ell}) \ge \ell_{th} \wedge c > \mu, \text{or} \\ \max(\bar{\ell}, \hat{\ell}) < \ell_{th} \wedge c < \mu \end{cases} \\ c & \text{otherwise} \end{cases}$$

Take-aways from cluster level

- *Capacity*: sustainable throughput with <1% SYN drop
  - Ensured client-perceived performance
  - Easy to measure and control
  - Three different estimation methods: off-line, CPU-based, and queue-monitoring.
- Queue-monitoring yielded superior results
  - Applicable to diverse and changing workloads
  - Low-overhead – no need to look for lower-overhead solutions
- Energy-conservation
  - Up to 44% energy savings with simple capacity-based controller
  - No prior knowledge is required
  - Short start-up and shutdown delays benefit energy conservation

Start-up and Shutdown Delay

- Long start-up and shutdown delay reduced the benefits of cluster resizing.



Single Server—tougher problem

- Consists of CPU(s), memory, disks, and network interfaces
- How to coordinate usage of these components to reduce energy consumption and meet application requirements?
  - Low-hanging fruits: do this for each component in isolation
  - High-hanging (unreachable) fruits: do this for all components cooperatively.

Low-hanging Fruit 1: CPU

How to reduce CPU's power consumption?

- Hardware:
  - Limit parallelism and speculative execution
  - Improve circuit technology
- Software:
  - Perform fewer computations
  - Improve algorithms and mechanisms

Use only as much of CPU as needed

- Example: real-time DVS (RT-DVS)
  - DVS algorithms that maintain RT guarantees
  - Simple enough for online scheduling
  - Work closely with existing RT sched algs.

Max bang out of fixed engery budget

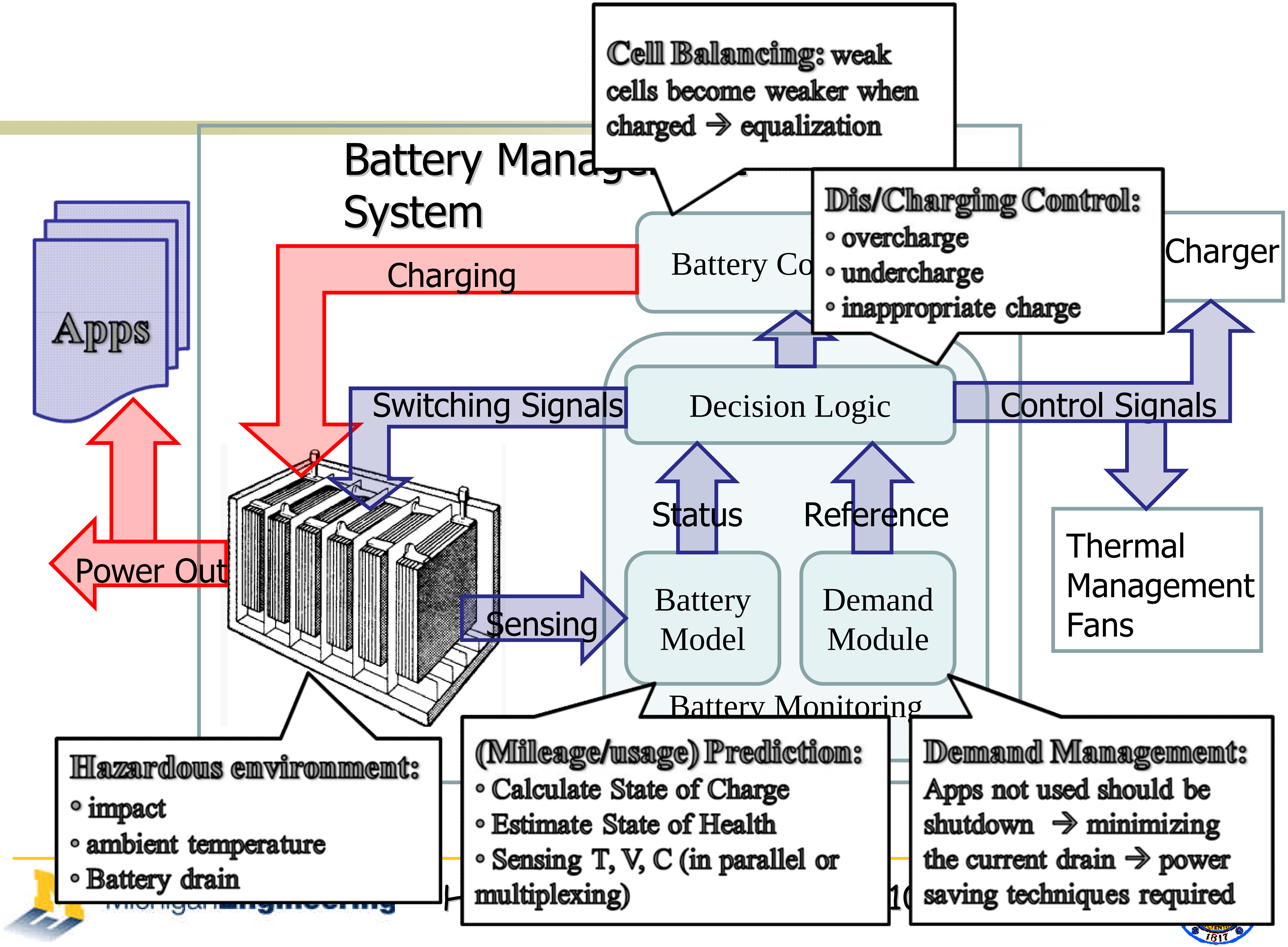
- Many technologies and techniques to save energy, but few guidelines on how to use different techniques to make best of the limited energy resource
- Energy-aware QoS (EQoS) formulates this into a tractable, optimal-selection problem:
  - Leveraging halt and DVS techniques
  - Task and QoS adaptation
  - Meeting system runtime goals
  - Maximizing benefits/utility

LH Fruit 2: Power-Aware Virtual Memory

- Modern memory devices operate in multiple power levels at different latencies.
- Elevate the decision making of managing power-state of modules onto an OS level.
- Apply NUMA concepts in a novel way
  - Use a NUMA management layer to abstract physical memory allocation to reduce the power dissipation of memory on a per-process basis.
  - Use advanced techniques such as shared library aggregation and page migration to further reduce the power dissipation of memory.

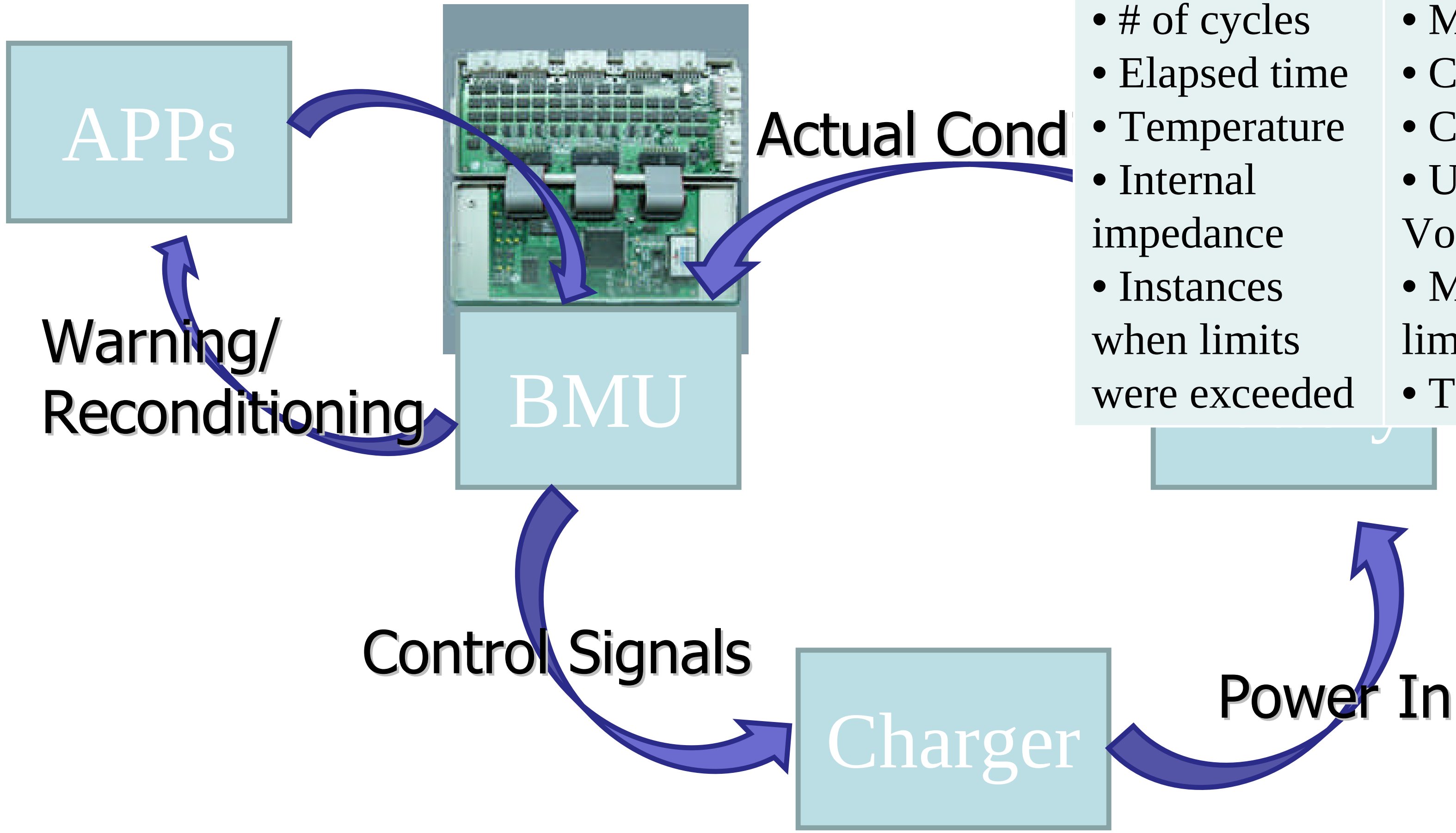
LH Fruit 3: Utilizing Free Disk Space

- Un-utilized disk space is abundant but a waste!
- What should we do with free disk space?
- FS2 [SOSP'06], an efficient solution:
  - Store replicas of poorly-placed disk blocks in free space
  - Replicas can be accessed in addition to their original
- What are the benefits of replication?
  - Improves performance
  - Reduces energy-consumption



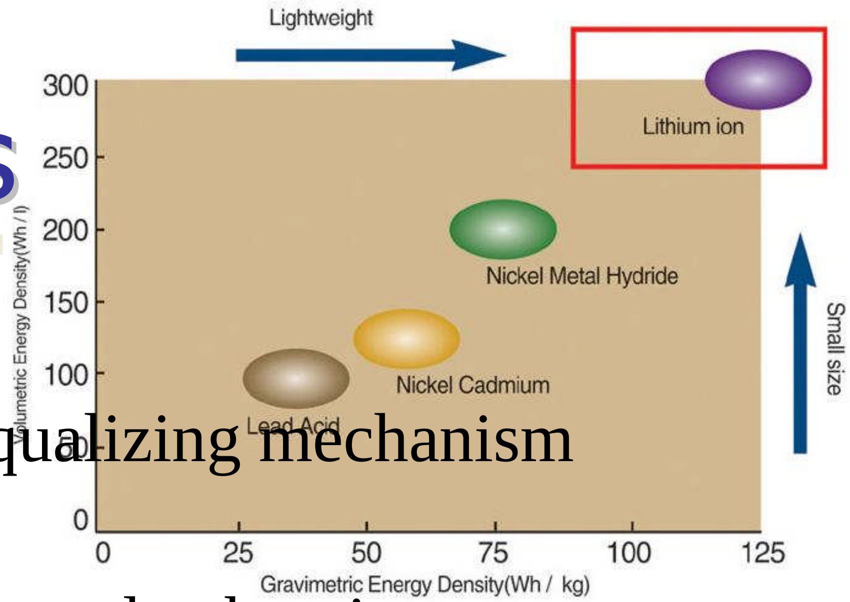
Battery Management System
Cell Balancing: weak cells become weaker when charged → equalization
Dis/Charging Control:
∘ overcharge
∘ undercharge
∘ inappropriate charge
Apps
Charging
Battery Co
Charger
Switching Signals
Decision Logic
Control Signals
Status
Reference
Thermal Management Fans
Power Out
Sensing
Battery Model
Demand Module
Battery Monitoring
Hazardous environment:
∘ impact
∘ ambient temperature
∘ Battery drain
(Mileage/usage) Prediction:
∘ Calculate State of Charge
∘ Estimate State of Health
∘ Sensing T, V, C (in parallel or multiplexing)
Demand Management:
Apps not used should be shutdown → minimizing the current drain → power saving techniques required

Control Flow: Feedback Loop



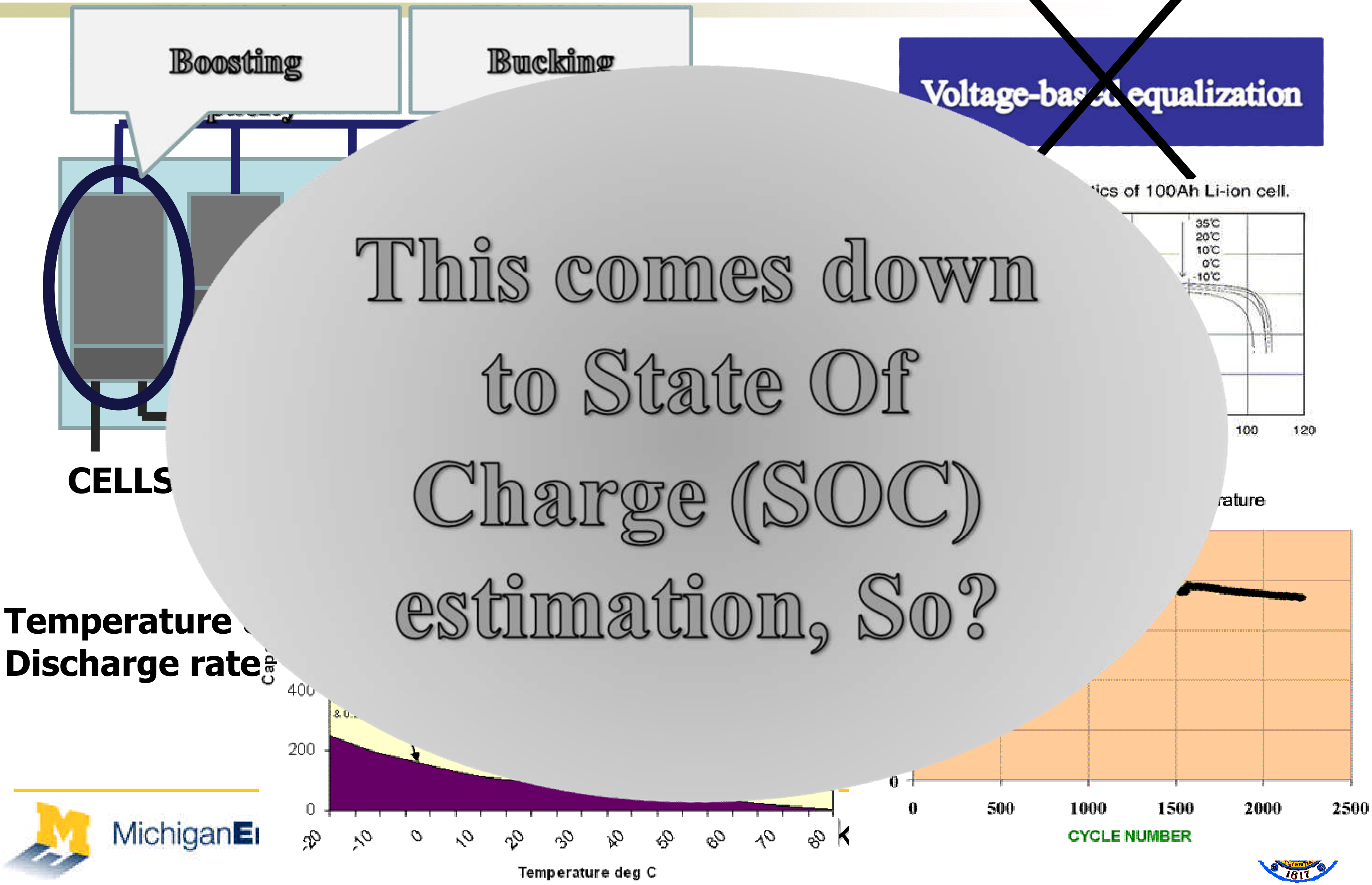
Dynamic info	Static info
• # of cycles • Elapsed time • Temperature • Internal impedance • Instances when limits were exceeded	• Manu. data • Cell chemistry • Cell capacity • Upper & Lower Voltage limits • Max current limits • Temp limit

Research Problems



- Microscopic mechanisms:
  - Cell balancing: cell-charge equalizing mechanism
  - Diagnosis of state of health
  - Prevention of overcharging & undercharging
- Macroscopic mechanisms:
  - Monitoring → Diagnostics → Detection → Reconfiguration → Prevention → Monitoring
- Battery Model: Prediction of how many miles the vehicle may run, state of health, state of charge, and driving patterns

Dominant Factors in Battery Failure



This comes down to State Of Charge (SOC) estimation, So?

Conclusions

- Energy is precious for economic and environmental reasons
- So, use it only when and as much as you need, especially for large datacenters, vehicles, homes & buildings
- The lifetime of energy source can be extended by conserving its usage as well as monitoring and controlling it.