Distributed Pervasive Systems

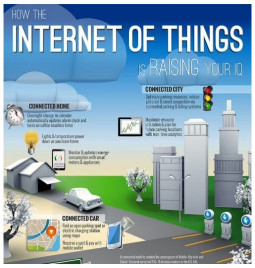
- Distributed Pervasive Systems
- Sensor Networks
- Energy in Distributed Systems (Green Computing)
  - Smart building case study

Pervasive Computing

- Computing become pervasive or ubiquitous
- Computing everywhere
- Rise of “smart” devices
  - smart cities, smart homes, smart highways, smart classroom, ...

Distributed Pervasive Computing

- Internet of things
  - ability to network devices and have them communicate
- Sensor networks
  - Large networks of sensors
- Driven by miniaturization of computing
  - Tiny sensors with computing and communication capability



Example Applications

- Smart home

Personal Health Monitoring

- Sensors to monitor fitness, diabetes, blood pressure, detect falls

Google tests prototype of diabetes-tracking 'smart' contact lens

Typical Smart Apps Architecture

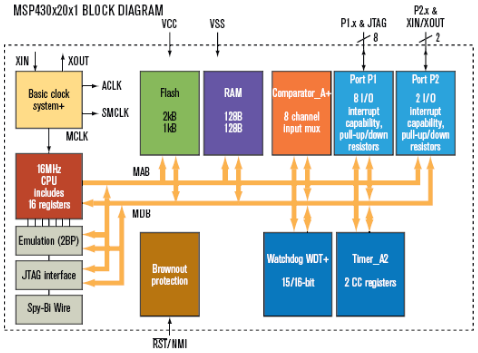
- Personal device to mobile phone to the cloud
  - Upload data to cloud via a mobile device (or directly)
  - Low-power communication to phone
  - Cloud provides analytics and provides feedback to phone
- Environmental sensors to internet to the cloud
  - Internet-enabled sensors
  - direct upload to servers / cloud
  - Cloud provides analytics and provides dashboard

Sensor Platform

- Smart devices are a sensor platform
- Resource-constrained distributed system
- Typical Sensor platform
  - Low-power radios for communication
    - 10-200kbit/sec
  - Small CPUs
    - E.g. 8bit, 4k RAM.
  - Flash storage
  - Sensors
  - Battery driven or self-powered

Small CPUs

- Example: Atmel AVR
  - 8 bit
  - 4 KB RAM
  - 128 KB code flash
  - ~2 MIPS @ 8MHz
  - ~8 mA
- Example: TI MSP430
  - 16 bit (sort of)
  - 10 KB RAM
  - 48 KB code flash
  - 2 mA

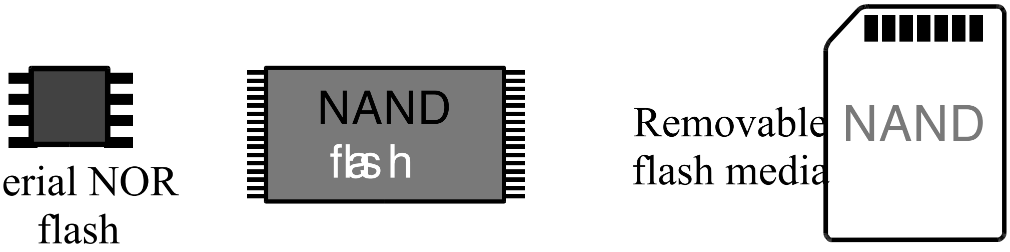


Higher-powered processors:
ARM7 (Yale XYZ platform)
32 bit, 50 MHz, >>1MB RAM
ARM9 (StarGate, others)
32 bit, 400 MHz, >>16MB RAM

Low Power Radios

- ISM band – 430, 900, or 2400 MHz
- Varying modulation and protocol:
  - Custom (FSK?) – Mica2, 20 kbit/s
  - Bluetooth, Zwave, Treads
  - Zigbee (802.15.4) - ~200kbit/sec
- Short range
  - Typically <100 meters
- Low power. E.g. Chipcon CC2420:
  - 9-17 mA transmit (depending on output level)
  - 19 mA receive
- Listening can take more energy than transmitting

Flash Storage



- Raw flash
- Small (serial NOR), very low power (NAND)
- Page-at-a-time write
- No overwrite without erasing
- Divided into pages and erase blocks
- Typical values: 512B pages,
  32 pages in erase block
- Garbage collection needed to gather free pages for erasing

“Cooked” flash

Disk-like interface
512B re-writable blocks
Very convenient
Higher power consumption

Battery Power

- Example: Mica2 “mote”
- Total battery capacity: 2500mAH (2 AA cells)
- System consumption: 25 mA (CPU and radio on)
- Lifetime: 100 hours (4 days)
-
- Alternatives:
- Bigger batteries
- Solar/wind/… (“energy harvesting”)
- Duty cycling

Sensors

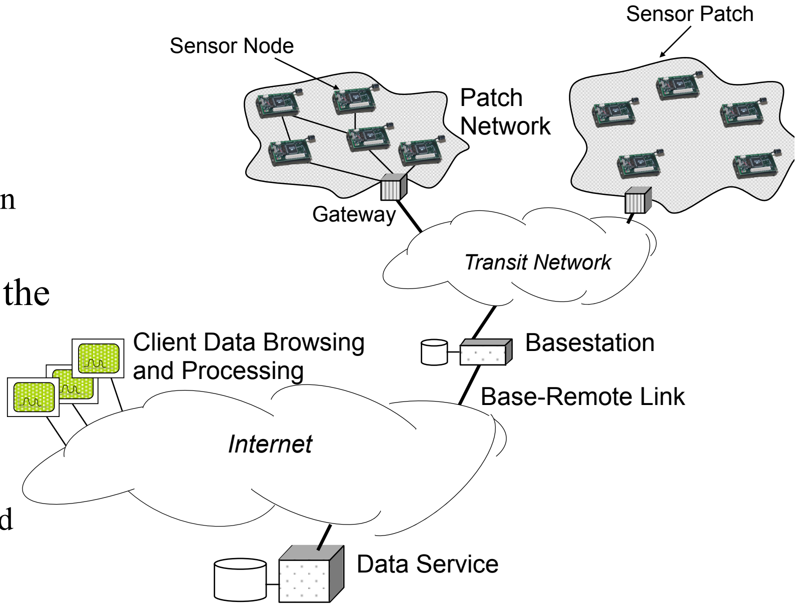
- Temperature
- Humidity
- Magnetometer
- Vibration
- Acoustic
- Light
- Motion (e.g. passive IR)
- Imaging (cameras)
- Ultrasonic ranging
- GPS
- Lots of others…

Self-harvesting Sensors

- Harvest energy from environment to power themselves
  - tiny solar panels, use vibration, airflow, or wireless energy

Typical Design Issues

- Single node
  - Battery power or how to harvest energy to maximize lifetime
- Inside a network of sensors
  - Data aggregation
  - Duty cycling
  - Localization, Synchronization
  - Routing
- Once data is brought out of the network (server-side processing)
  - “Big data” analytics
  - Derive insights
  - Make recommendations, send alerts
  - Provide active control



Green Computing

- Greening of computing
  - Sustainable IT
  - How to design energy-efficient hardware, software and systems?

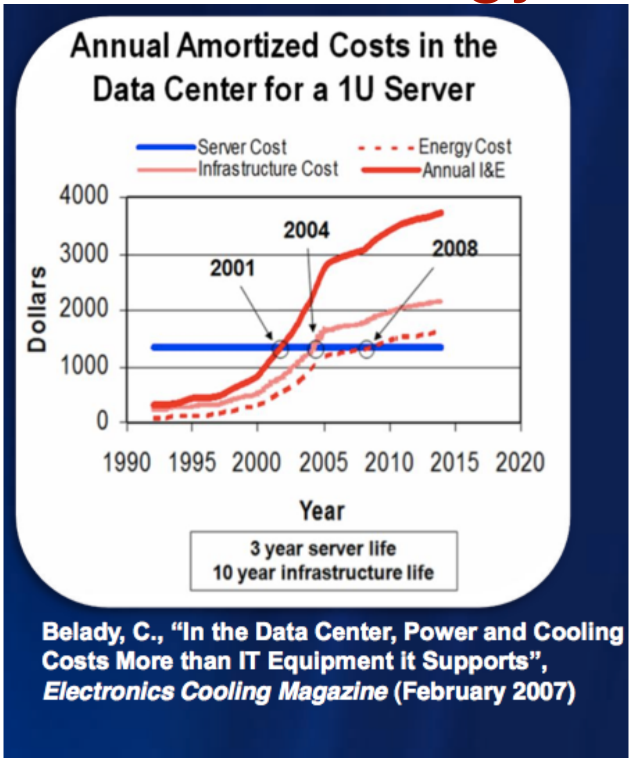
- Computing for Greening
  - Use of IT to make physical infrastructure efficient?
    - Homes, offices, buildings, transportation

Some History

- Energy-efficient mobile devices a long standing problem
  - Motivation: better battery life, not green

- Recent growth of data centers
  - More energy-efficient server design
  - Motivation: lower electricity bills
    - Green systems, lower carbon footprint
- Apply "Greening" to other systems
  - IT for Greening

Data Center Energy Costs



Belady, C., "In the Data Center, Power and Cooling Costs More than IT Equipment it Supports", *Electronics Cooling Magazine* (February 2007)

Energy Bill of a Google Data center

- Assume 100,000 servers
- Monthly cost of 1 server
  - 500W server
  - Cost=(Watts X Hours / 1000) * cost per KWH
  - Always-on server monthly cost = $50
- Monthly bill for 100K servers = $5M
- What about cost of cooling?
  - Use PUE (power usage efficiency)
  - PUE =2 => cost doubles
  - Google PUE of 1.2 => 20% extra on 5M (~ $6M)

How to design green data centers?

- A green data center will
  - Reduce the cost of running servers
  - Cut cooling costs
  - Employ green best practices for infrastructure

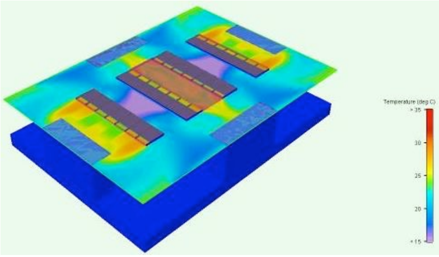
Reducing server energy cost

- Buy / design energy-efficient servers
  - Better hardware, better power supplies
  - DC is more energy-efficient than AC
- Manage your servers better!
  - Intelligent power management
  - Turn off servers when not in use
  - Virtualization => can move apps around

Reducing cooling costs

- Better air conditioning
  - Thermal engineering / better airflow
  - Move work to cooler regions

- Newer cooling
  - Naturally cooled data centers
    - Direct air cooling

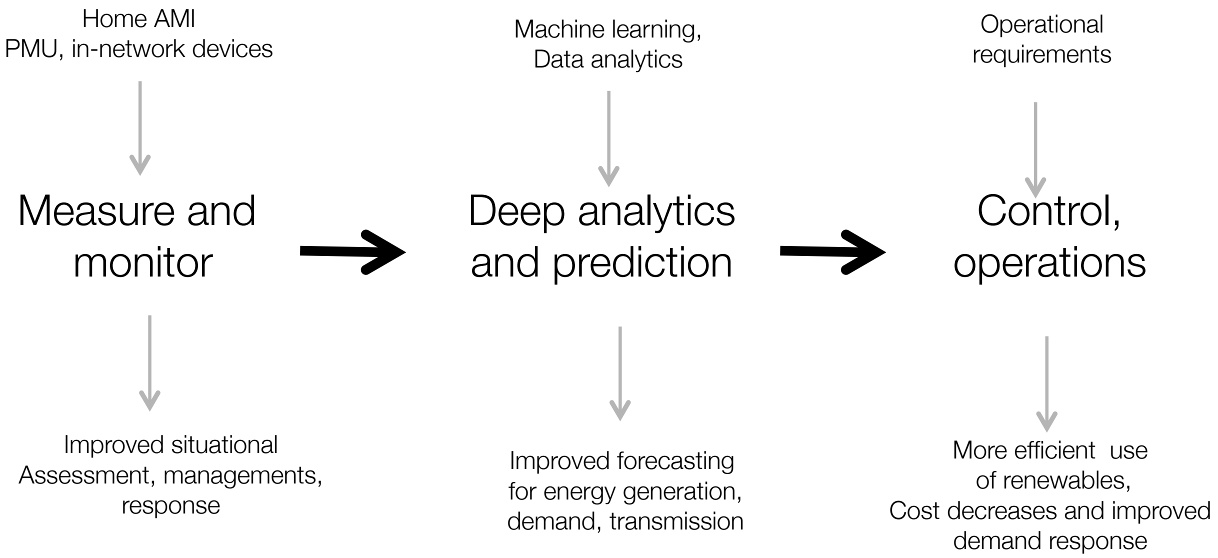


IT for Greening

- **Case Study: Smart Buildings**
- **Building as an example of a distributed system**
  - Distributed pervasive system
  - Sensors monitor energy, occupancy, temperature etc
  - Analyze data
  - Exercise control
    - switch of lights or turn down heat in unoccupied zones
  - Use renewables to reduce carbon footprint

- How can we use IT to make buildings green?
  - Use sensors, smart software, smart appliances, smart meters …..

Approach



Home AMI
PMU, in-network devices

Measure and monitor

Improved situational
Assessment, managements,
response

Machine learning,
Data analytics

Deep analytics and prediction

Improved forecasting
for energy generation,
demand, transmission

Operational
requirements

Control, operations

More efficient use
of renewables,
Cost decreases and improved
demand response

Potential Solution

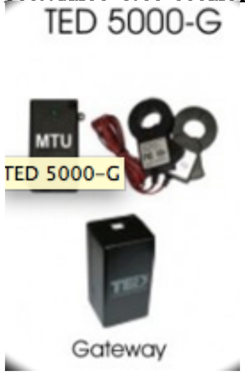
- Monitor and profile usage
  - Power supply/demand profile
- Increase Efficiency
  - Turn on/off systems automatically
  - Consolidate computers
  - Tune various subsystems
- Use Alternative Energy Sources
  - Tune systems to variable energy supplies

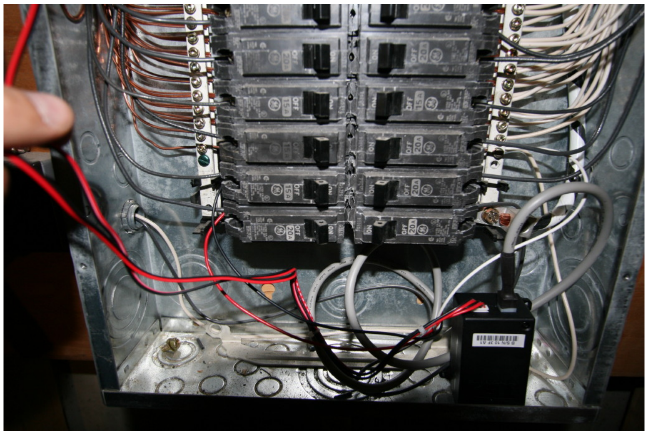
Outlet level Building Monitoring

- Designed sensors for power outlet monitoring
  - Based on the Kill-A-Watt design
- Modified sensor with low-power wireless radio
  - Transmits data to strategically placed receivers
  - Use plug computers for receivers

Meter level Monitoring

- Install on main panel





Analyzing the data

Energy monitors / sensors provide real-time usage data

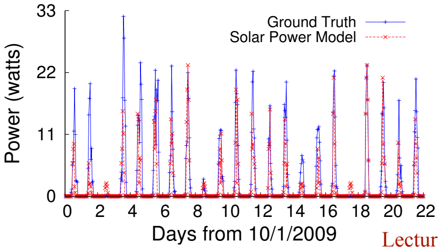
- Smart meters:
- Building monitoring systems (BMS) data from office / commercial buildings

Modeling, Analytics and Prediction

- Use statistical techniques, machine learning and modeling to gain deep insights
  - Which homes have inefficient furnaces, heaters, dryers? Are you wasting energy in your home?
  - Is an office building's AC schedule aligned with occupancy patterns?
  - When will the aggregate load or transmission load peak?

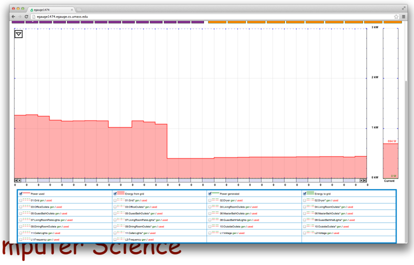
Use Renewables

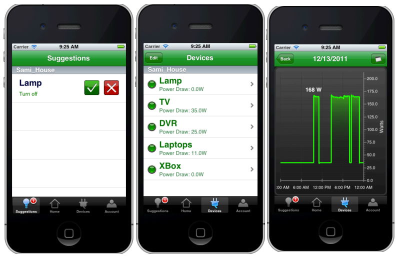
- Rooftop Solar, Solar Thermal (to heat water)
- Design predictive analytics to model and forecast energy generation from renewables
  - Use machine learning and NWS weather forecasts to predict solar and wind generation
- Benefits: Better forecasts of near-term generation; "Sunny load" scheduling



People: Feedback and Incentives

- How to exploit big data to motivate consumers to be more energy efficient?
  - What incentives work across different demographics?
- Analysis can reveal insights into usage patterns, waste, efficiency opportunities
  - Smart phone as an engagement tool to deliver big data insights to end-users
    - Provide highly personalized recommendations, solicit user inputs, motivate users





Course Summary

- Architectures
- Distributed communication
- Processes, Scheduling, Virtualization
- Naming
- Canonical Problems
- Replication, Consistency, Fault Tolerance
- Distributed File Systems
- Security in Distributed Systems
- Distributed middleware
- Special topics: streaming, web, cloud, pervasive, green