Experiences with IoT and
The New Internet as a Platform of Things

Rich Wolski
Computer Science Department
University of California, Santa Barbara
October 19, 2020
Internet of Things

Projecting the ‘Things’ Behind the Internet of Things
From 2014-2020, IoT grows at an annual compound rate of 23.1% CAGR

Internet of Things - number of connected devices
2015 to 2025 (in billions)

1 trillion objects expected to connect to the Internet by 2025

Globalization and technology are changing the face of the business world

Adoption of new technologies is accelerating
Time to reach 50 million users, years

Technological breakthroughs are speeding up
The path toward mobile Internet

Disruptive Technologies to Watch

Mobile Internet
$4 trillion–$11 trillion

Automation of knowledge work
$5 trillion–$7 trillion

Internet of Things
$3 trillion–$6 trillion

McKinsey Global Institute


Including consumer surplus, McKinsey Global Institute, Disruptive Technologies, May 2013
Three Observations About Putting Things on The Internet

• Power is everything

• Deployment is “soft”-ware and must be engineered

• Build “repurpose built” instead of “purpose built.”
The Internet is really The Datacenter(s)
Power is Everything in this Setting

- Power is more expensive to provision than compute, communication, storage
  - For example: ~$20K to put solar in for 4 PC-class edge systems and long-range WiFi
- Power infrastructure is fragile
  - Dust, sun angle, industrial accidents
- Power infrastructure is invasive
  - Can’t trench conduit through the space
  - Solar towers interfere with farming operations
- Scaling properties are terrible
  - More power means more expensive, fragile, and invasive

*If every “Thing” is to go on the Internet, we need to engineer for power efficiency.*
  - It is infrastructure and not just battery life
What’s Wrong with this Picture?
The Device is in the Wrong Place

- The sensors were designed for deployment in rocky soil, with all-day sun, at altitude
- Can’t measure carbon sequestration in a charcoal pit
- $250K in equipment, 2 person days of installation
- Two separate contracts
  - One for the equipment and software development
  - One for the “siting” and deployment
- Abandoned 18 months after installation
What Went Wrong?

• The software development was considered completely separate from how and where the device was to be deployed.

• Agile development for IoT must include deployment
  – Where the device goes influences the software which influences where the device goes which...
  – *The Deployment Manager missed the scrum*

• *For IoT, Deployment is Software*
What Went Right?

- They donated the equipment to our research group (Thanks!)
  - Built for long-term deployment (still up and running)

- Needs to be reprogrammed for a different purpose
  - Current sensor readings are useless

- The “rip and replace” model for devices is infeasible

- Repurposing the device is hard
  - Three data loggers
  - Primitive networking
  - Totally insecure
  - When we did repurpose, not enough power!

- Build “repurpose” built instead of ”purpose built.”
The New World (circa 2025)

• Power is everything
• Deployment is software
• Build repurpose built instead of purpose built
• Conclusion (surprisingly):

The Internet, at present, is engineered “backwards” from what is required for IoT
Cloud Today = E-Commerce & Entertainment:
Data Moves From Cloud to User Devices

- Download (read) dominated: streaming entertainment/content

Public Clouds/CDNs:
- Amazon Web Services
- YouTube
- Netflix
- Microsoft Azure

UCSB
Cloud Today

• Serving consumer content is “read dominated”
  – The “writes” are web page clicks or, at worst, single uploads

• Aggregating click activity generates “Big Data”

• Inference and prediction analytics (Big Stats) “work” because the sample sizes are huge

• Goal: Make an inference or a prediction about a specific individual
  – Big Data and Big Stats are needed because people are so diverse
Upload (write) dominated: sensing and monitoring.

‘N’ Billion IoT Devices by 2025

Public Clouds
- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
Not Big Data but Big Little Data

• Data sets and stats are not that big

• *Rich’s IoT Data Conjecture*
  – The actionable relevance of IoT data decays like the square of the distance from where it is gathered

• Not Big Data but lots and lots of Little Data

• Data is often read only once (to create an aggregate) which makes the data even smaller.
  – Why move the data half way across the country to use it in an aggregate and throw it away?

• *Rich’s Data Motto*
  – The more the data moves, the more expensive, power-using, and failure prone the system is.
Why is IoT Hard?
Why are Cloud Applications Built as Services?

- Services *react* to user activity
  - Run only when needed => *efficient*

- Services hide the infrastructure detail
  - APIs allow the “back ends” to be deployed automatically => *programmable deployment*

- Services are reprogrammable
  - Extensions or code modifications happen behind the service interface => *repurposable*

- Services are abstract, hiding and managing heterogeneity
  - REST and HTTP(S) make functionality *portable*
Ergo, for IoT, Devices Should Run Services

- **Power efficient**: services only respond when they are prompted
- **Programmable Deployment**: cloud services often use deployment software
- **Repurposeable**: Devices are multi-purpose and fully programmable
- **Portable**: Devices are massively heterogeneous calling for a common abstraction
- **Idea**: *Flip the Internet*
  - Devices run services
  - Edge runs services
  - Cloud runs applications as clients and user interfaces
Devices-as-Services

- Need a service development, deployment, and hosting environment that is multi-scale
  - Common set of familiar abstractions for taming heterogeneity
  - Deployment support: same service (unmodified) can run on device, or edge, or cloud
  - Repurpose support: as devices or edge are repurposed, extended, or replace, services change and possible move

- Our Approach
  - Portable, multi-scale, distributed “Functions as a Service” (FaaS)
Cloud Functions as a Service

- Also called “serverless” computing
  - AWS Lambda, Google Functions, Azure Functions
- For cloud
  - Developer writes “stateless” functions in high-level language
  - Packaged with libraries and dependencies
  - Uploaded to FaaS platform in a specific cloud
  - Executed by the cloud in a “container” (no servers required for isolation)
  - Event-driven programming model
    - Functions are only triggered when a specific event occurs (much cheaper!)
  - Storage and communication are through external cloud services
CSPOT

• **Serverless Platform Of Things** (first version in C)

• Multi-scale, distributed FaaS
  – Run on microcontrollers, small Linux (Raspberry Pi, Intel NUC), Campus Clouds (NSF Aristotle), public clouds (AWS, Azure, Google, IBM)
  – CSPOT program ports (source code level) without modification between scales

• CSPOT includes
  – Append-only storage abstraction (distributed data durability, eventual consistency)
  – Log-based runtime that tracks causal dependencies (debugging)

• **CSPOT is a way to write multi-scale network-facing services, microservices, and “nanoservices”**

• Open Source
  – [https://github.com/MAYHEM-Lab/cspot.git](https://github.com/MAYHEM-Lab/cspot.git)
  – Join us! Contribute!
UCSB IoT using CSPOT

Device Tier

Edge Tier

Regional Tier

Public Cloud Tier

Service Distribution Network

On Farm

Public Clouds

 Courtesy Chandra Krintz
UCSB Computer Science
Portable and Power Efficient

- Benchmark dispatch times for clock read FaaS Handlers
- CSPOT is 1 to 2 orders of magnitude faster than AWS Lambda
- Power being proportional to latency

<table>
<thead>
<tr>
<th>System</th>
<th>Mean (ms)</th>
<th>Stdev (ms)</th>
<th>95% (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Esp8266 ucontroller</td>
<td>38</td>
<td>0.15</td>
<td>40</td>
</tr>
<tr>
<td>Raspberry Pi (ARM)</td>
<td>37</td>
<td>6.8</td>
<td>48</td>
</tr>
<tr>
<td>Intel NUC</td>
<td>4.0</td>
<td>0.63</td>
<td>4.9</td>
</tr>
<tr>
<td>UCSB Cloud</td>
<td>5.0</td>
<td>1.6</td>
<td>7.0</td>
</tr>
<tr>
<td>AWS SPOT EC2 in C</td>
<td>5.0</td>
<td>0.96</td>
<td>6.6</td>
</tr>
<tr>
<td>AWS SPOT EC2 in Python</td>
<td>18</td>
<td>3.1</td>
<td>23</td>
</tr>
<tr>
<td><strong>AWS Lambda</strong></td>
<td><strong>253</strong></td>
<td><strong>90</strong></td>
<td><strong>584</strong></td>
</tr>
</tbody>
</table>
Edge and Cloud

- **Device**: CC3220SF (80 MHz, 256KB, HW Crypt, WiFi)
- **Edge**: Intel NUC8I7HNK (3.1 GHz, 32 GB, WiFi)
- UCSB Lab WiFi and campus networking
- Pub/sub streaming telemetry application

<table>
<thead>
<tr>
<th>System</th>
<th>Time (ms)</th>
<th>Energy (mj)</th>
<th>Code (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device-&gt;Edge CSPOT</td>
<td>119</td>
<td>22</td>
<td>48</td>
</tr>
<tr>
<td>Device-&gt;Azure IoT Edge</td>
<td>1715</td>
<td>251</td>
<td>75</td>
</tr>
<tr>
<td>Device-&gt;Cloud CSPOT</td>
<td>165</td>
<td>32</td>
<td>51</td>
</tr>
<tr>
<td>Device-&gt;Azure IoT Hub</td>
<td>3168</td>
<td>457</td>
<td>130</td>
</tr>
</tbody>
</table>
What Will the Brave New World Need?

- Taming heterogeneity will require *one (efficient) ring to rule them all*
  - CSPOT is source-code level programmatically deployable
    - We are working on making deployment equivalently efficient
- Distributed and secure registration and discovery
  - Blockchain?
- High-level platform programming abstractions
  - CSPOT is an assembly language for distributed IoT
  - The Internet using Apache and HTTP
IoT, AI, and Future Cloud

• High-performance Computing (HPC) is really going to matter
  – Predictions about the physical world based on sensor telemetry requires faster-than-
    real time simulations
  – *Future of AI for IoT is a fusion of neural networks, physical models, and HPC simulations*

• Tiered cloud architecture
  – The data will move only as far from its source as it needs to and no farther
  – Service Distribution Network: like a CDN but for services

• Energy efficiency research meets Computer Science
  – UCSB Institute for Energy Efficiency is a venue for studying these problems “end-to-
    end.”
Talent Recruiting: The New Resume’
Thanks!

- Collaborators: UCSB, UCSB IEE, LREC, CalPoly, Fresno State, Powwow Energy, Sedgwick Reserve, Private Growers

- Support: Google, Intel, IBM Research, Microsoft Research, NSF, NIH, California Energy Commission

Students:

Fatih Bakir
Alexis Cole
Kerem Celik
Raymond Deng
Gareth George

Shereen Hussein
Wei-Tsung Lin
Nazmus Saquib
Michael Zhang

UCSB RACELab
The Lab for Research on Adaptive Computing Environments
Computer Science Department, Harold Frank Hall (E-5), Santa Barbara, CA

rich@cs.ucsb.edu, ckrintz@cs.ucsb.edu