2018 Spring Retreat
2018 Spring Retreat:
Lab Overview and Update

John Ousterhout
Faculty Director
Thank You, Sponsors!
Platform Lab Faculty

- **Bill Dally**: Architecture
- **Sachin Katti**: Networking
- **Christos Kozyrakis**: Architecture, System Software
- **Phil Levis**: Embedded Systems
- **Nick McKeown**: Networking
- **John Ousterhout**: Granular Computing (Fac. Director)
- **Guru Parulkar**: Networking (Exec. Director)
- **Balaji Prabhakar**: Networking
- **Mendel Rosenblum**: Distributed Systems, Networking
- **Keith Winston**: Networking, Granular Apps
- **Matei Zaharia**: Big Data, Cloud Computing
Lab mission:
define new hardware/software platforms
that enable exciting new classes of applications
What is a Platform?

- General-purpose hardware or software substrate
- Simplifies construction of a class of applications (or higher-level platforms)
  - Solves common problems
  - Usually introduces (simplifying) restrictions
- Examples:
  - Relational databases → Enterprise applications
  - HTTP + HTML + JavaScript → Internet commerce
  - GFS + MapReduce → Big Data (large-scale analytics)
  - Smart phones + GPS → Google Maps, Uber, …
Computers
Manage Information

Computers
Control Devices

Collaborative Device Swarms

The next frontier in computing
Big Control

- Hypothesis: device swarms will be managed centrally from datacenters
- Infrastructure for Big Data can’t meet needs
- Need new platforms to enable large-scale control applications

### Application Platform for Swarm Control
- Scalable abstractions for control

### Granular Computing
- Enable applications composed of microsecond-scale tasks

### Self-Programming Networks
- Predictable and adaptable communication
Why Granular Computing?

Centralized control of large device swarms

1000 – 1,000,000 devices

Real-time response

1 – 1000 events/sec/device

Control Application

Cascade of internal tasks:
- Integration
- Fusion
- Inference
- Condition checks
- Triggers
- Planning
- Actuation
Why Granular Computing?

New model for cloud computing

- Traditional approach: run large jobs for a long time
  - Hadoop
  - Spark
- Rent servers by the hour
- New approach: smaller execution units (serverless)
  - Execute individual requests (function calls)
  - Pay for CPU time and memory used
  - Example: Amazon Lambda
- Minimum size still large: 100 ms

Can we support 1000x smaller units of execution? What is the limit?
Why Granular Computing?

Real-time data-intensive processing

- **Goals:**
  - Incorporate large amounts of data in decision-making
  - Respond in real-time: < 200 ms

- **Not possible today**
  - Big Data applications execute for minutes → hours

- **Solution: granular computing**
  - Divide computation into large numbers of small tasks
  - Run tasks concurrently
Problem: Software Stacks too Slow

- Existing software stacks highly layered
  - ✔ Great for software structuring
  - ✗ Bad for latency

- Designed for disk-based computing (10 ms+)

- Today’s stacks can’t support granular computing:
  - Linux thread create/delete: 9 µs
  - Linux RPC: 20 µs best case
    - 200 – 1000 µs if network loaded
  - Schedule Spark job: 1 second?

Granular computing requires new software stack
Projects in Granular Computing

- **System infrastructure:**
  - Homa network protocol: low latency datacenter transport
  - Arachne: core-aware thread management
  - NanoLog: extremely fast logging
  - gg: framework for granular apps on serverless
  - Pocket: ephemeral storage for gg
  - Shinjuku: preemptive run-time system for low-latency applications

- **Applications:**
  - ExCamera: granular video processing
  - Millisort: sorting at scale in 1 ms

*Overall goal: enable smallest possible tasks (10-100 µs?)*
Self-Programming Networks

- **New approach to network management:** the network programs itself
  - Operators specify declarative goals
  - Network monitors, controls itself to achieve goals
  - Leverage machine learning techniques

- **Why SPNs?**
  - Writing control programs very difficult today
  - Challenges increasing over time:
    - Diverse applications with different needs
    - Unpredictable new applications and workloads
    - Increasing requirements for performance and predictability
    - Multiple modalities: mobile, long-haul wired, datacenter
Self-Programming Networks, cont’d

- **Control specified at a higher level**
  - What you want (declarative)
  - Not how to get it

- **Network monitors, controls itself:**
  - **Sense** to collect data on behavior, needs
  - **Infer** global state, bottlenecks, causes of symptoms
  - **Learn** best control behaviors (e.g. deep reinforcement learning)
  - **Forecast** future behaviors, impact of changes
  - **Control** network configuration, policies

- **Heavy use of machine learning**
SPN Projects

- Clock synchronization
  - Large-scale trials
  - Use at higher levels (e.g. consensus, DBs, distributed ledgers)
- Network telemetry via tomography (also in trials)
- ChatBot: querying and visualization of SPNs
- Smart NICs (push SPN mechanism to NICs)
- Self-driving radios
- Decoupling prediction and control
- New pipeline for mapping data (new approach to compression)
New Initiatives

- Smart NICs project (Prabhakar, Rosenblum)
- Accelerator for SAT (Dally)
- Massively parallel lambda computing (Kozyrakis, Winstein, Zaharia)
- Baby steps towards a platform for granular computing (Ousterhout)
- ORAN Alliance (Katti)
  - Standardize 5G network architecture and interfaces
  - Technical chair: Sachin Katti
Awards

- **Test of Time:**

- **Best papers:**
  - Ana Klimovic, et al: Memorable Paper Award at NVM Workshop
  - Turakhia, et al.: Best Paper Award at ASPLOS 2018 (Darwin)
Recent/Imminent Graduates

Manu Bansal  
Techniques for building Predictable Stream Processing Pipelines

Mingyu Gao  
Near-Data Processing

Yilong Geng  
Self-Programming Networks: Architecture and Algorithms

Sam Grossman  
Graph Processing

Amit Levy  
Multiprogramming a 64 KB Computer Safely and Efficiently

Raghu Prabhakar  
Coarse-Grain Reconfigurable Architectures

Zi Yin  
Natural Language Processing for ChatBot
Thursday Agenda

1:00  Welcome, Introductions, Platform Lab Overview  
      John Ousterhout

1:45  SDR: Self Driving Radios  
      Sachin Katti

2:10  The Case for Decoupling Prediction and Control in Modern Networks  
      Sandeep Chinchali

2:35  Update on Arachne  
      Henry Qin

3:00  Break

3:30  DIY Hosting for Online Privacy  
      Shoumik Palkar

3:55  Accelerating Data Analytics with FPGAs  
      James Thomas

4:20  Pocket: Ephemeral Storage for Serverless Analytics  
      Ana Klimovic

4:45  Lightning talks for posters

5:15  Reception/posters

6:30  Dinner

8:00  Evening activities
Friday Agenda

9:00  Closing the Loop on Secure Operating System Design  
      Phil Levis

9:25  Teams of Collaborating Robots for Flexible Manufacturing  
      Mac Schwager

9:50  Shinjuku: Reconciling Low Tail Latency with Preemptive Scheduling  
      Kostis Kaffes

10:15 Wedge: A New Frontier for Pull-Based Graph Processing  
      Sam Grossman

10:40  Recreation and Informal Conversations

12:00  Lunch

1:00  Salsify: Low-Latency Video Through Joint Control of a 
      Video Codec and Transport Protocol  
      Keith Winston

1:25  Millisort: An Experiment in Granular Computing  
      Seo Jin Park

1:50  X-Ray Vision using Wireless Signals  
      Manikantku Kotaru

2:15  Pantheon: a Community Evaluation Platform for Congestion Control  
      Francis Yan

2:40  Break

3:00  Progress on the SPN ChatBot: Handling Compound and 
      Domain-Specific Queries  
      Zi Yin and  
      Vin Sachidananda

3:25  Byzantine Fault Tolerant Clock Synch. in Datacenter Networks  
      Shiyu Liu

3:50  Industrial Feedback
Questions/Discussion