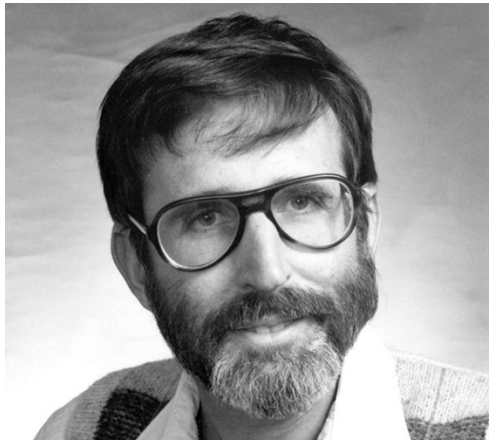
Confessions of a Systems Researcher

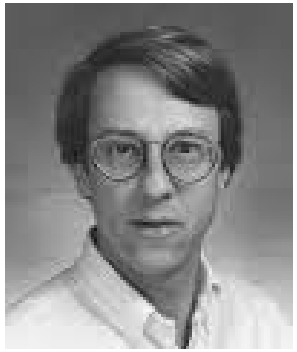
Kishore Ramachandran

My mentors at Wisconsin



Larry Landweber
Networking

Marvin Solomon
OS, Networking,
PL/Compilers

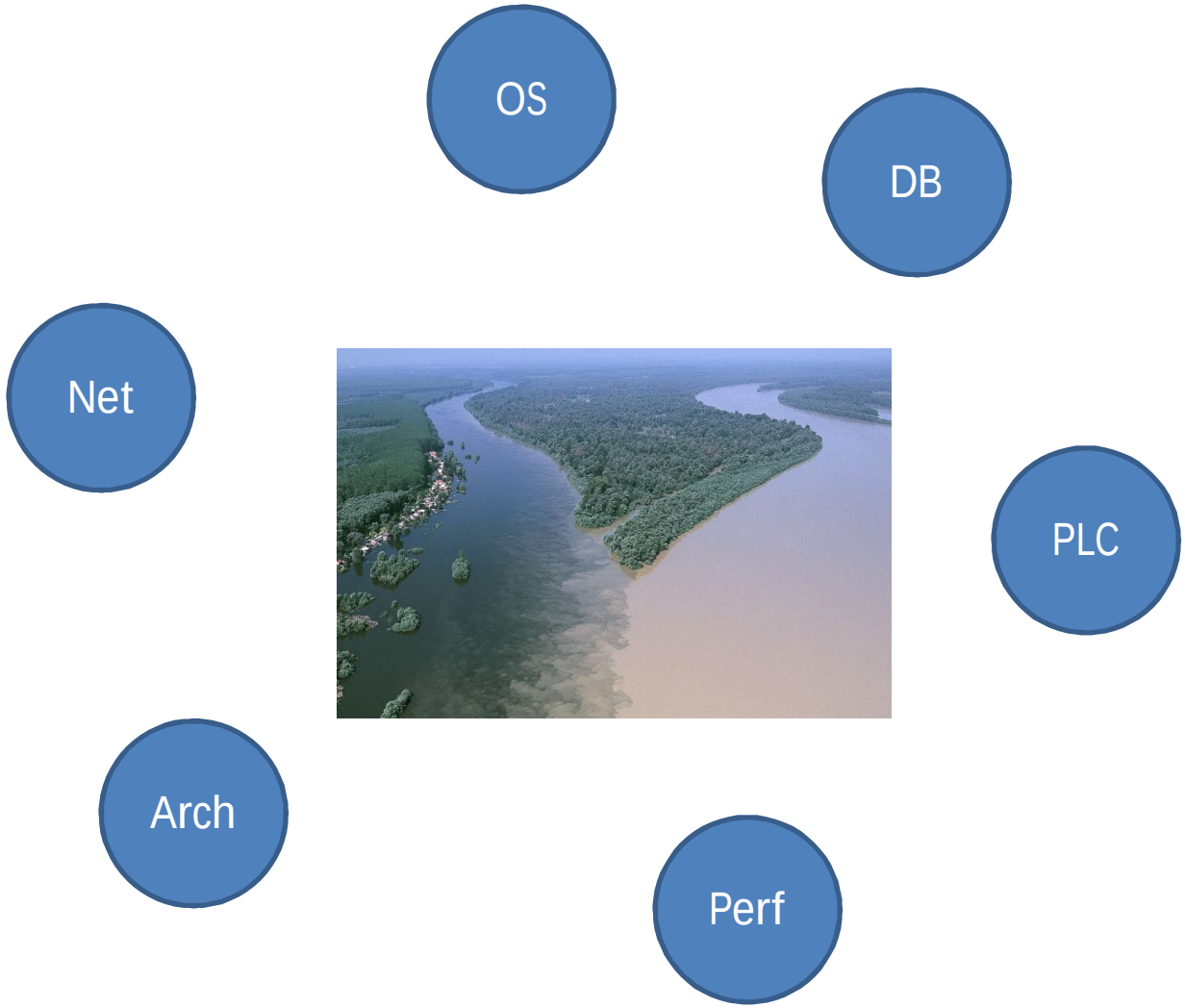


Dave Dewitt
DB

Charlie Fischer
PL/Compilers

Jim Goodman
Architecture

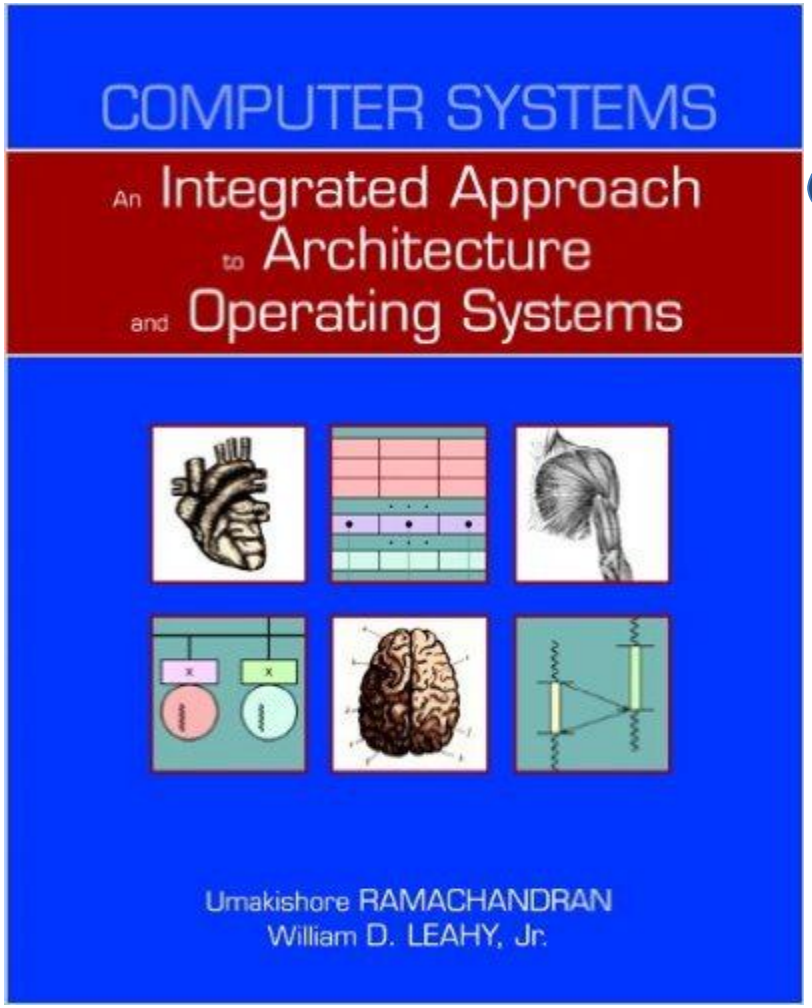
Mary Vernon
Perf. Modeling



PhD Thesis: Hardware Support for Interprocess Communication

OS courses

Architecture courses



VLSI course

Sensor Nets

Pervasive Computing


Sensor Networks
Data Engineering
Performance Modeling
Architecture
Parallel Processing
Even Theory!
Distributed Systems

Recipe for Disaster!!

- Don’t belong to any one community
- No one may know you enough

Advice:

Don’t do what I did!!

- Honestly, I will do exactly the same thing if I have to do it all over again!
- I was having a lot of fun
- Graduated an eclectic bunch of PhDs (27 in all)
  - OS types
  - Architecture types
  - Performance types
  - Database types

Jeff Ullman

Alan Demers (1975)

Marvin Solomon (1977)

M. V. S. Ramanath (1981) U. Western Ontario
Nancy Jarrell (1982) IBM Cambridge
Will E. Leland (1982) Bellcore
Mary D. P. Leland (1983) Lucent/Bell Labs
** The above two people are married
Jonathan Kepecs (1984) Legato Network Systems
Prasun Dewan (1986) U. North Carolina
HongHai Shen (1994) IBM Santa Teresa
William Kalsow (1986)

Umakishore Ramachandran (1986)

Mustaque Ahamad

great grand uncle!

Yousef Khalidi (1989) SUN Microsystems
Joonwon Lee (1991) KAIST, Korea
Number of grandchildren – unknown
Walter B. Ligon III (1992) Clemson Univ.
Number of grandchildren – 6+
Martin Davis (1992) SYSTRAN Inc.
Ajay Mohindra (1993) IBM Watson
Anand Sivasubramaniam (1995) Penn State
Number of grandchildren – 17+
Vibby Gottemukkala (1995) IBM Watson
Sreenivas Gukal (1995) Informix
Gautam Shah (1995) IBM Kingston
Aman Singla (1997) SGI
Zack Kurmas (2004) Grand Valley State
Sameer Adhikari (2004) Intel
Arnab Paul (2005) Intel
Matt Wolenetz (2005) Microsoft
Josh Fryman (2005) Intel
Rajnish Kumar (2006) Georgia Tech
Nissim Harel (2006) Jambool
Xiang Song (2008) Microsoft
Hasnain A. Mandviwala (2008) Ask.com
David B. Hilley (2009) Google
Nova Ahmed (2010) Georgia Tech
Bikash Agarwalla (2010) Facebook
Hyojun Kim (2012) IBM Almaden
Mungyung Ryu (2014) Google
Kirak Hong (2014) Google
Lateef Yusuf (2014) Amazon
Dave Lillethun (2015) Georgia Tech

(Full tree here…: http://infolab.stanford.edu/~ullman/pub/jdutree.txt)

A variety of partners in crime!

In short:

Do what I did...

At your own peril!!

Evolution of Systems Research (borrowing heavily from “History Day”)

- First, factoids for 60’s, 70’s, 80’s, 90’s, and 2000’s
- Defining moments that changed the landscape of computing

60's: Beginnings of systems community

- In the beginning there was only one "systems" conference
  - SOSP started in 1967
    - No boundary between OS, architecture, and programming
    - Everyone got together to brainstorm about how to build better systems (hardware and system software)
    - Idea of networking computers together was just getting born
- Computer Structures textbooks
  - Language oriented architecture
  - Application oriented architectures
  - Always connecting software and hardware

  (Bell and Newell: Computing Structures, 1971)

60's: Seminal System Concepts Emerged impacting hardware/software

- Multiprogramming (mid 60's in IBM 360)
- Time sharing (MIT CTSS, 1962)
  - Virtualizing the hardware for co-memory resident programs
- Quest for location independent addressing (early 50's)
  - Gave birth to virtual memory (early 60's)
- Principle of locality
  - Gave birth to caching (early 60's)
- Quest for one single parameter to tune the system performance
  - Gave birth to the working set concept (Denning, 1966)
- Microprogramming (Wilkes, 1958)
  - Structured systematic approach to hardware design
- Taxonomy for parallel machines (Flynn, 1966)

70's: Conference silos and the software/hardware disconnect

- Conference silos started with architecture and PL
  - ISCA started in 1974
  - POPL started in 1974
- Programming evolved into programming language design to aid application development
- Electrical engineers were responsible for architecture evolution
- OS was building abstractions to hide the hardware details
  - As an aside, "virtualization" ideas date back to the beginning of computer systems building
- ISA kept evolving limited only by the wild imagination of the hardware designer and available real estate on PCB!
- Increasing disconnect between hardware and software
- Every new processor needed a complete new system software design and implementation

70's: Heydays of CISC and dawn of Unix

- Semiconductor memories (MOS)
- Microprogramming paved the way for CISC
  - more control memory => more complex instructions
  - Byte-string instructions in IBM mainframes
- MOS paved the way for microprocessor revolution
  - Intel i432 (way ahead of its time…capability-based architecture)
  - Intel 8086
- Dataflow architecture (Dennis, 1974)
  - Seminal idea got into system design (OS, PLC) though never flourished as a way of implementing hardware architecture
- While architecture was getting bulkier OS was getting thinner
  - MIT Multics leads to Bell Labs' Unix (1969-1970)
- Lots more happening on the system software side…more on that in a minute…

80's: Tighter integration of hardware and system software

- Emer and Clark at DEC (80's)
  - 20% of VAX instructions -> 60% of microcode, but only account for 0.2% of execution time!

  => Shift towards hardware folks talking to software folks again...
- Compiler-driven instruction-set design
  - Gave birth to RISC (early 80's)
- CMOS revolution
  - Gave birth to "Killer Micros"
- New conferences fostering this integration
  - ASPLOS, PACT, NSDI, SenSys, Embedded systems
- Traditional silos reinventing themselves and/or making course corrections
  - Sigmetrics, Micro, ISCA, HPCA, SOSP, OSDI, PLDI

80's and 90's: Heydays of parallelism

- Multiprocessors built out of Killer Micros
  - Shared memory
    - Memory hierarchies for multiprocessors
      - Memory consistency models
      - Cache coherence protocols
    - NUMA architectures
  - Message passing
    - Exotic interconnects
      - FAT Tree
      - Butterfly
- OS mechanisms for multiprocessors
  - Threads and Events
  - Synchronization
  - Scheduling
- Programming constructs
  - Led to POSIX standards (Pthreads library)
- Applications that drove the marketplace
  - Technical computing

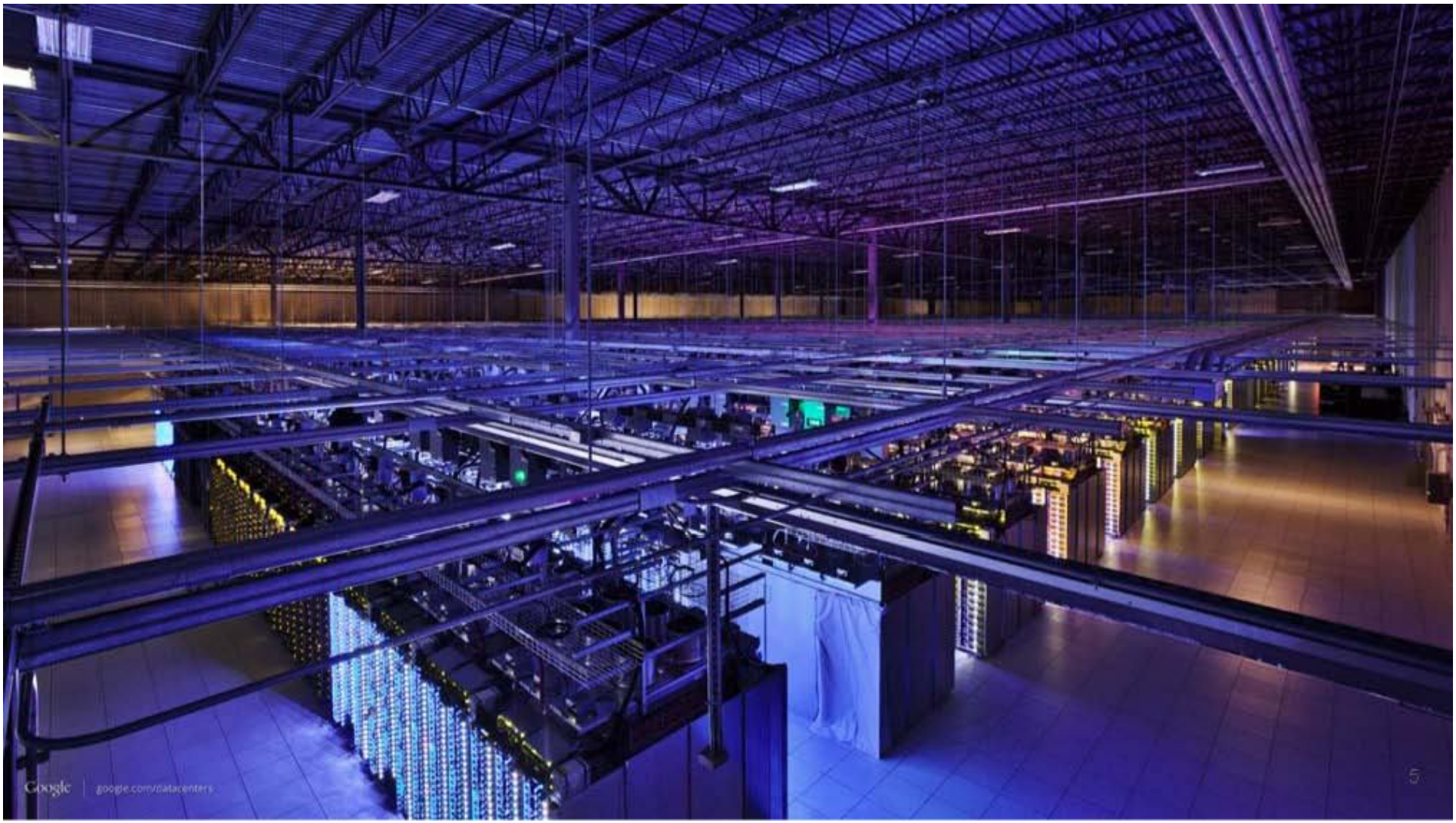
2000's: Era of clusters

- Application landscape changed
  - Web searches
  - Indexing web content
  - Serving interactive queries
  - Social networks
- Need for large (upwards of 1000s of CPUs)
  - Birth of scale out computing
  - Computing as a utility
- Autonomic computing

Google Data Center

Google data center: Courtesy Jeff Dean's slide deck from History day at SOSP 2015

Google Data Center



Google data center: Courtesy Jeff Dean's slide deck from History day at SOSP 2015

Defining moments in computing systems evolution

System software/Networking

- FORTRAN (Backus, 1954)
- Multics (Corbato, 1965)
- Arpanet IMP (1969)
- Arrival of LAN (mid 70's)
- C (Ritchie, 1970)
- Unix (Ritchie and Thompson, 1969)
- NFS (Sun microsystems, 1984)
- Linux (Torvalds, late 80's)

Hardware/Architecture

- Microprogramming (Wilkes, 1958)
- IBM 360 (Early 60's)
- Emer and Clark at DEC (80's)
  - 20% of VAX instructions -> 60% of microcode, but only account for 0.2% of execution time!

  => Shift towards hardware guys talking to software guys again...
- Killer micros and the CMOS revolution (late 70's and 80's)
- IBM 801 (Cocke, 1980)
- RISC 1 (Patterson, 1982)
- VLIW (Fisher, 1984)

Defining moments in computing systems evolution

System software/networking

- Network Protocol stack as part of OS (Berkeley Unix - 1980's, Wisconsin IBM Project – 1980s)
- WWW (Berners-Lee, 1989)
- Microkernel-based OS design (90's)
- Rebirth of Virtualization (2000s)
- Internet of Things (Now!)

Hardware/Architecture

- Memory consistency models and Multiprocessor cache coherence (80's and early 90's)
- Scalable multiprocessors (90's)
  - Stanford DASH , SGI Origin, KSR-1 and KSR-2, TMC CM-5
- Embedded systems (2000's)
- Multicore and Manycores (2000's)
- Scale out clusters (2000's)
  - IBM Blue Gene, Google cluster
- Mobile + Cloud (Now!)

Now what keeps me awake these days?

Embedded Pervasive Lab

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Pervasive side of the house

- Embedded devices treated as black boxes
- Two flagship projects
  - Transient Social Networks
  - Mobile Cloud
- Both relevant to IoT research theme

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Transient Social Networks

- Encounter-based social networks formed on-demand to support collaboration
- Four attributes
  - Participants on mobile devices
  - Spatially local (application dependent spatial reach)
  - Temporally local (with access to historic events)
  - Common shared interests

Example Applications

Stadium Vendor

Local Auctions

Mobile Gaming

Emergency response

Village Market

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Research Issues

- Membership management in the presence of network- and location-induced churns
- Autonomic configuration
- Content distribution and discovery
- Reliable messaging among the members of TSN
- Energy and bandwidth efficiency
- Location privacy/trust

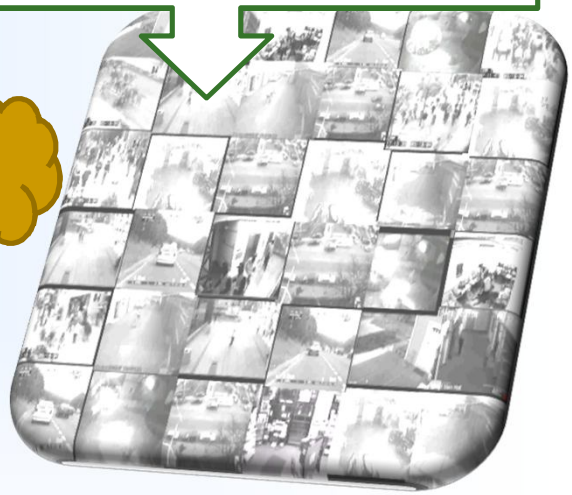
Mobile Cloud: Marrying Smart Devices and Utility Computing

Sporting Events

Ji-sung Park's goal

Surveillance

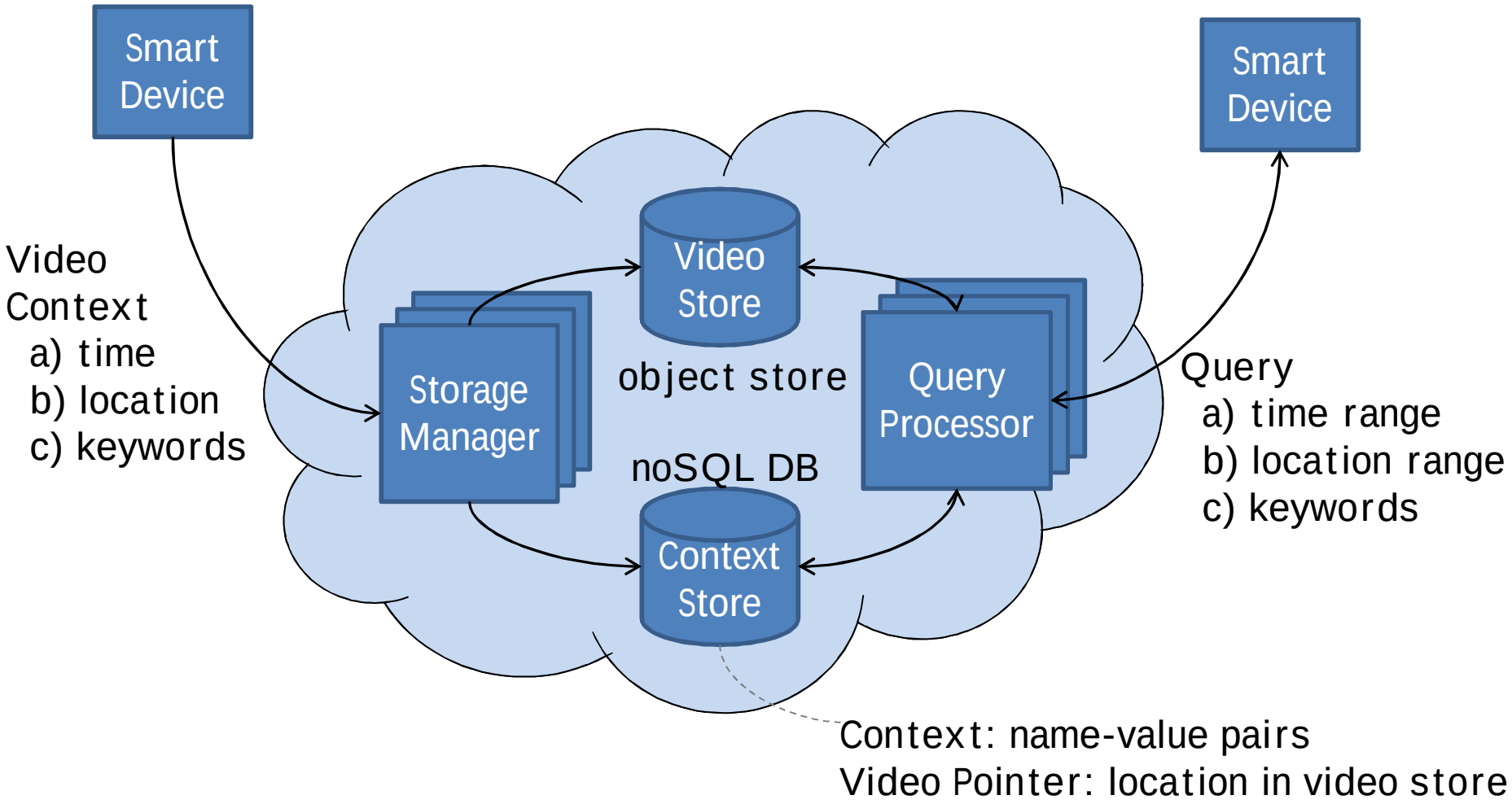
Show me only those cameras with a child in view



Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Mobile Cloud Architecture



Research Issues

- Re-architecting current Cloud for latency-sensitive applications
  - Extend utility computing model to the edge of the network
- Develop an architecture for the Mobile Cloud
  - Sensor registration and discovery system
  - Intelligent filtering at the edge
  - Programming model & run time system
  - Scalable streaming pub/sub system
  - Efficient distributed resource management

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Embedded side of the house

What can we do to improve software architecture of embedded platforms?

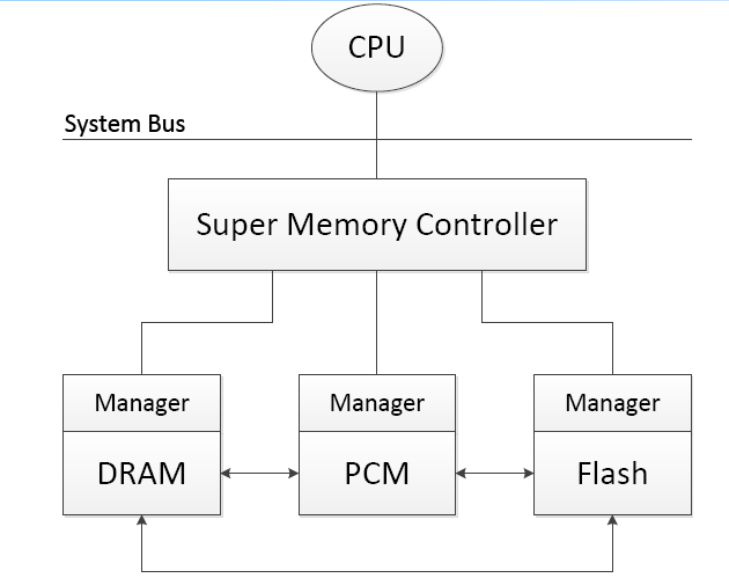
Four flagship projects

- OS Structure with infinite Heterogeneous memory
- Storage architecture of HTTP streaming server
- Informed storage management of smart phones
- Memory virtualization for smart devices

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

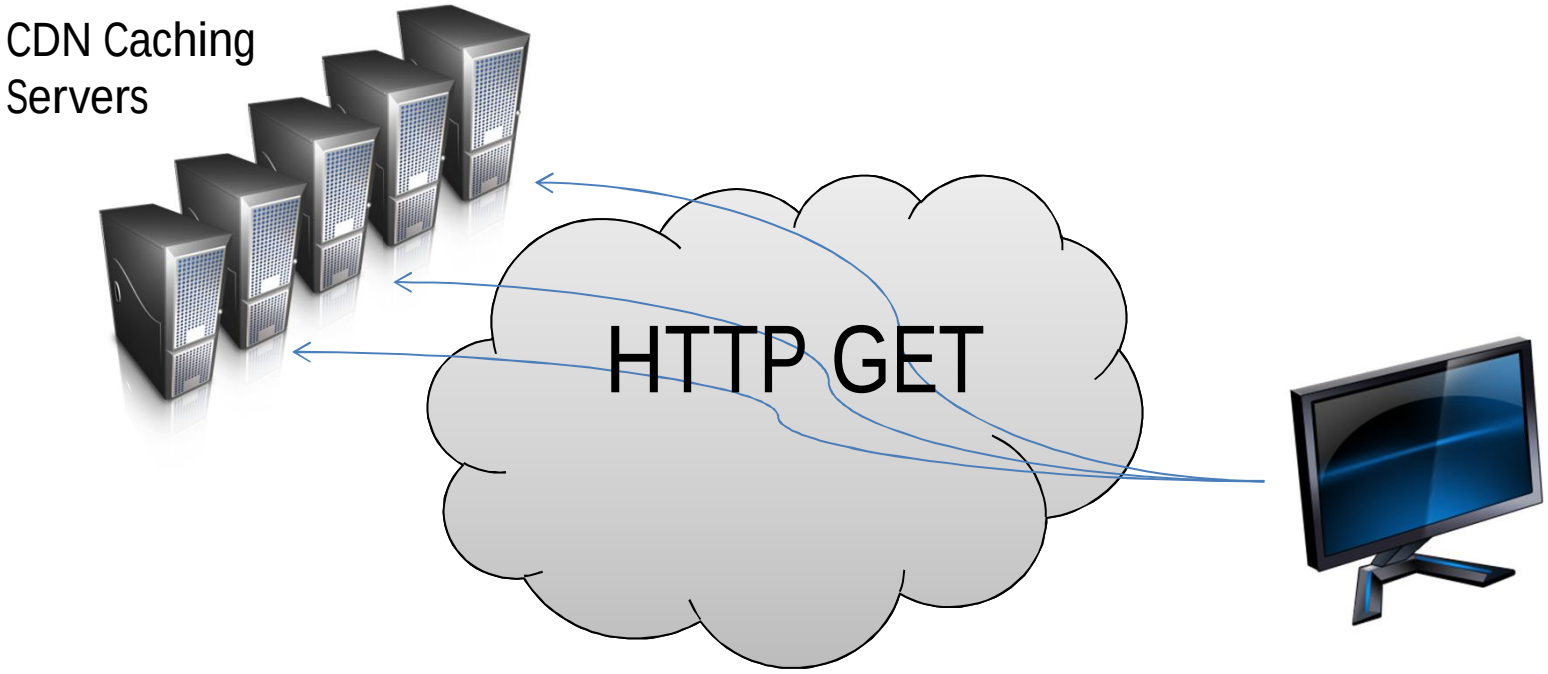
Heterogeneous Memory



- Rethink OS structure
  - Process address space mapping
    - Eliminate page cache?
  - File system
    - Non-blocking I/O?
  - Power management
  - Hypervisor design
    - Memory balancing and de-duplication
- Applies to server side and client side system software

Adaptive HTTP Streaming

- Streaming by WEB servers
- Client adaptively selects different bitrate segments
- NETFLIX



Research issues

- Multi-tier (DRAM, Flash, Disk) storage server design considerations for HTTP DASH streaming
- RAID-ed flash-based storage server design
- Caching policy in flash
- Managing the read/write I/O requests to the storage

Informed Storage Management for Smart Phones

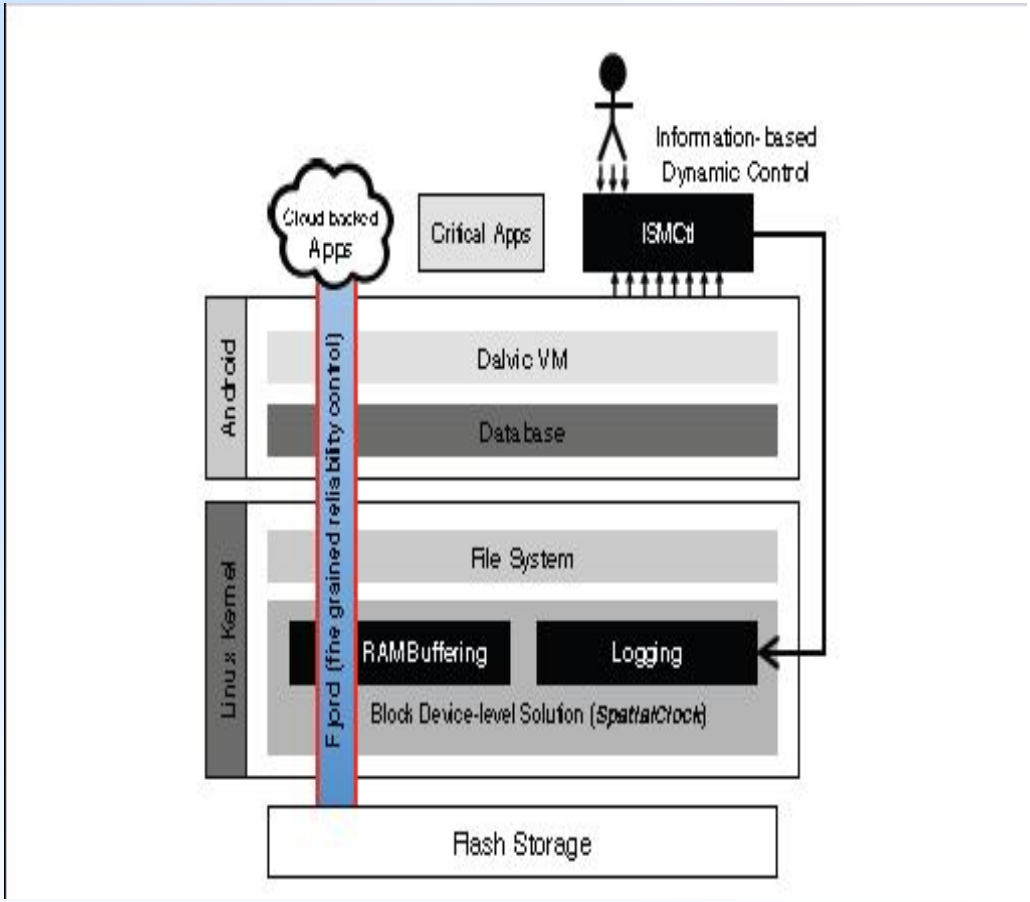
- Flash storage exhibits very different performance characteristics relative to the traditional HDD
- But, current operating systems are not engineered to support flash storage adequately.
- Consequently, flash storage is the Achilles' heel when it comes to performance of mobile platforms!
- Besides, it is very difficult to improve storage performance without losing reliability.

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Research Issues

- How to gracefully combine Logging and RAM Buffering solutions?
- How to selectively relax reliability of cloud-backed Applications?
- How to exploit system wide information in smartphones?



Georgia Tech College of Computing

STAR Samsung Tech Advanced Research Center

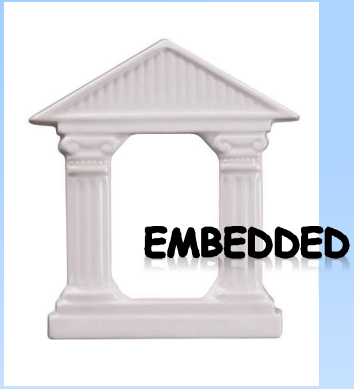
Memory Virtualization for Smart Devices

- Why?
  - Data centers
    - Accountability
    - Performance
  - Smartphones and IoT
    - Versatility
    - Memory is finite (as opposed to CPU, I/O Cycles)
    - Amount of allocated memory has direct impact on the performance of a VM
    - Memory need of VMs varies over time
    - Ensuring integrity of critical apps

Research Issues

- Coordinated memory management across VMs
- Figures of merit
  - Latency
  - Scheduler interference
  - Memory Fragmentation

Recap



- OS Structure with infinite Heterogeneous memory
- Low-Cost High-Performance HTTP Video Streaming
- Informed Storage Management for Smartphones
- Memory Virtualization for Smart Devices

- System Support for Transient Social Networks
- Mobile Cloud: Marrying Smart Devices and Utility Computing

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Current graduate inmates!

Dushmanta Mohapatra, Wonhee Cho,

Yeonju Jeong, Steffen Maas, Thomas Wolf, Enrique Saurez Apuy

Recently escaped!

Hyojun Kim (IBM then Startup),

Lateef Yusuf (Amazon then Google),

Mungyung Ryu (Google),

Kirak Hong (Google),

Dave Lillethun (GT)

Plus a number of MS and UGs

Georgia Tech College of Computing

STAR
Samsung Tech Advanced Research Center

Concluding remarks

- Still confused but in a good way!
  - Lots of things to work on
  - Not enough time in the day
- Next time I give this talk I want your picture in my “partners in crime” slide!!