Internet-scale Computing:
The Berkeley RADLab Perspective

Randy H. Katz
randy@cs.berkeley.edu
28 May 2007
Rise of the Internet DC

• Observation: Internet systems complex, fragile, manually managed, evolving rapidly
  – To scale Ebay, must build Ebay-sized company
  – To scale YouTube, get acquired by a Google-sized company

• Mission: Enable a single person to create, evolve, and operate the next-generation IT service
  – “The Fortune 1 Million” by enabling rapid innovation

• Approach: Create core technology spanning systems, networking, and machine learning

• Focus: Making datacenter easier to manage to enable one person to Analyze, Deploy, Operate a scalable IT service
• Microsoft and Google race to build next-gen DCs
  – Microsoft announces a $550 million DC in TX
  – Google confirm plans for a $600 million site in NC
  – Google two more DCs in SC; may cost another $950 million -- about 150,000 computers each

• Internet DCs are the next computing platform

• Power availability drives deployment decisions
Datacenter is the Computer

- Google *program* == Web search, Gmail,…
- Google *computer* == Warehouse-sized facilities and workloads likely more common

Luiz Barroso’s talk at RAD Lab 12/11/06

Sun Project Blackbox 10/17/06

Compose datacenter from 20 ft. containers!
- Power/cooling for 200 KW
- External taps for electricity, network, cold water
- 250 Servers, 7 TB DRAM, or 1.5 PB disk in 2006
- 20% energy savings
- 1/10th? cost of a building
Declarative Datacenter

- **Synthesis**: change DC via written specification
  - DC Spec Language compiled to logical configuration
- **OS**: allocate, monitor, adjust during operation
  - Director using machine learning, Drivers send commands
• **$S^2ML$ Strengths**
  – Handle SW churn: Train vs. write the logic
  – Beyond queuing models: Learns how to handle/make policy between steady states
  – Beyond control theory: Coping with complex cost functions
  – Discovery: Finding trends, needles in data haystack
  – Exploit cheap processing advances: fast enough to run online

• **$S^2ML$ as an integral component of DC OS**
Datacenter Monitoring

• $S^2$ML needs data to analyze
• DC components come with sensors already
  – CPUs (performance counters)
  – Disks (SMART interface)
• Add sensors to software
  – Log files
  – D-trace for Solaris, Mac OS
• Trace 10K++ nodes within and between DCs
  – *Trace: App-oriented path recording framework
  – X-Trace: Cross-layer/-domain including network layer
Middleboxes in Today’s DC

- Middle boxes inserted on *physical* path
  - Policy via plumbing
  - Weakest link: 1 point of failure, bottleneck
  - Expensive to upgrade and introduce new functionality

- Policy-based Switching Layer: policy not plumbing to route classified packets to appropriate middlebox services
RIOT: RadLab Integrated Observation via Tracing Framework

- Trace connectivity of distributed components
  - Capture *causal connections* between requests/responses
- Cross-layer
  - Include network and middleware services such as IP and LDAP
- Cross-domain
  - Multiple datacenters, composed services, overlays, mash-ups
  - Control to individual administrative domains

- “Network path” sensor
  - Put individual requests/responses, at different network layers, in the context of an end-to-end request
DC Energy Conservation

• DCs limited by power
  – For each dollar spent on servers, add $0.48 (2005)/$0.71 (2010) for power/cooling
  – $26B spent to power and cool servers in 2005 grows to $45B in 2010

• Attractive application of S\textsuperscript{2}ML
  – Bringing processor resources on/off-line: Dynamic environment, complex cost function, measurement-driven decisions
    • Preserve 100% Service Level Agreements
    • Don’t hurt hardware reliability
    • Then conserve energy

• Conserve energy and improve reliability
  – MTTF: stress of on/off cycle vs. benefits of off-hours
• Within DC racks, network equipment often the “hottest” components in the hot spot
• Network opportunities for power reduction
  – Transition to higher speed interconnects (10 Gbs) at DC scales and densities
  – High function/high power assists embedded in network element (e.g., TCAMs)
M. K. Patterson, A. Pratt, P. Kumar,
“From UPS to Silicon: an end-to-end evaluation of datacenter efficiency”, Intel Corporation
DC Networking and Power

• Selectively power down ports/portions of net elements
• Enhanced power-awareness in the network stack
  – Power-aware routing and support for system virtualization
    • Support for datacenter “slice” power down and restart
  – Application and power-aware media access/control
    • Dynamic selection of full/half duplex
    • Directional asymmetry to save power, e.g., 10Gb/s send, 100Mb/s receive
  – Power-awareness in applications and protocols
    • Hard state (proxying), soft state (caching), protocol/data “streamlining” for power as well as b/w reduction
• Power implications for topology design
  – Tradeoffs in redundancy/high-availability vs. power consumption
  – VLANs support for power-aware system virtualization
Active Network Management

- Networks under stress: critical reliability problem in modern networks
- Technology for packet inspection is here
- Exploit for distributed network mgmt
  - Load balancing
  - Traffic shaping
Networks Under Stress

Vern Paxson, ICIR, "Measuring Adversaries"
“Background”
Radiation
--
Dominates traffic in many of today’s networks

= 596% growth/year

---
Vern Paxson, ICIR, “Measuring Adversaries”
Network Protection

• Internet robust to point problems like link and router failures ("fail stop")
• Successfully operates under a wide range of loading conditions and over diverse technologies
• 9/11/01: Internet worked well, under heavy traffic conditions and with some major facilities failures in Lower Manhattan
Network Protection

• Networks awash in illegitimate traffic: port scans, propagating worms, p2p file swapping
  – Legitimate traffic starved for bandwidth
  – Essential network services (e.g., DNS, NFS) compromised

• Need: active management of network services to achieve good performance and resilience even in the face of network stress
  – Self-aware network environment
  – Observing and responding to traffic changes
  – Sustaining the ability to control the network
• **Campus Network**
  – Unanticipated traffic renders the network unmanageable
  – DoS attacks, latest worm, newest file sharing protocol largely indistinguishable--surging traffic
  – In-band control is starved, making it difficult to manage and recover the network

• **Department Network**
  – Suspected DoS attack against DNS
  – Poorly implemented spam appliance overloads DNS
  – Difficult to access Web or mount file systems
Networks Failure

• Complex phenomenology
• Traffic surges break enterprise networks
• “Unexpected” traffic as deadly as high net utilization
  – *Cisco Express Forwarding*: random IP addresses --> flood route cache --> force traffic thru slow path --> high CPU utilization --> dropped router table updates
  – *Route Summarization*: powerful misconfigured peer overwhelms weaker peer with too many router table entries
  – *SNMP DoS attack*: overwhelm SNMP ports on routers
  – *DNS attack*: response-response loops in DNS queries generate traffic overload
Trends and Tools

• Integration of servers, storage, switching, and routing
  – Blade Servers, Stateful Routers, Inspection-and-Action Boxes (iBoxes)

• Packet flow manipulations at L4-L7
  – Inspection/segregation/accounting of traffic
  – Packet marking/annotating

• Building blocks for network protection
  – Pervasive observation and statistics collection
  – Analysis, model extraction, statistical correlation and causality testing
  – Actions for load balancing and traffic shaping

Load Balancing  Traffic Shaping
Generic Network Element

Input Ports

Buffers

Buffers

Buffers

“Tag” Mem

CP

Rules & Programs

Classification Processor

Interconnection Fabric

Action Processor

Output Ports
iBoxes implemented on commercial PNEs

- Don’t: route or implement (full) protocol stacks
- Do: protect routers and shield network services
  - Classify packets
  - Extract flows
  - Redirect traffic
  - Log, count, collect stats
  - Filter/shape traffic
Active Network Elements

- Server Edge
- Network Edge
- Device Edge

- NAT, Access Control
- Network-Device Configuration
- Firewall, IDS
- Traffic Shaper
- Server Load Balancing
- Storage Nets
Packeteer PacketShaper
Traffic monitor and shaper

Network Appliance NetCache
Localized content delivery platform

F5 Networks BIG-IP LoadBalancer
Web server load balancer

Ingrian i225
SSL offload appliance

Cisco SN 5420
IP-SAN storage gateway

Nortel Alteon Switched Firewall
CheckPoint firewall and L7 switch

NetScreen 500
Firewall and VPN

Extreme Networks SummitPx1
L2-L7 application switch

Cisco IDS 4250-XL
Intrusion detection system
Enterprise Network Architecture

Inspection-and-Action Boxes:
Deep multiprotocol packet inspection
No routing: observation & marking
Policing points: drop, fence, block
Observe-Analyze-Act

• Observe
  – Packet, path, protocol, service invocation statistical collection and sampling: frequencies, latencies, completion rates
  – Construct the collection infrastructure

• Analyze
  – Determine correlations among observations
  – “Normal” model discovery + anomaly detection
  – Exploit SLT

• Act
  – Experiment to test correlations
  – Prioritize and throttle
  – Mark and annotate
  – Control theory? Distributed analyses and actions
Observe-Analyze-Act

- Control exercised, traffic classified, resources allocated
- Statistics collection, prioritizing, shaping, blocking, ...
- Minimize/mitigate effects of attacks & traffic surges
- Classify traffic into good, bad, and ugly (suspicious)
  - Good: standing patterns and operator-tunable policies
  - Bad: evolves faster, harder to characterize
  - Ugly: cannot immediately be determined as good or bad
- Filter the bad, slow the suspicious, preserve the good
  - Sufficient to reduce false positives
  - Suspicious-looking good traffic slowed, but not blocked
Scenario

Server Edge

Distribution Tier

Internet Edge

Access Edge

PC

DNS

Spam Filter

FS

MS
Ops Problems Observed

• User visible services:
  – NFS mount operations time out
  – Web access also fails intermittently due to time outs

• Failure causes:
  – Independent or correlated failures?
  – Problem in access, server, or Internet edge?
  – File server failure?
  – Internet denial of service attack?
Network Dashboard

- Gentle rise in ingress b/w
- No unusual pattern
- Mail traffic growing
- Unusual step jump/DNS xact rates
- Decline in access edge b/w
Network Dashboard

CERT Advisory! DNS Attack!

- Gentle rise in ingress b/w
- Unusual step jump/DNS xact rates
- Decline in access edge b/w

FS CPU utilization
MS CPU utilization
b/w consumed
DNS CPU utilization

Mail traffic growing

In Web
Email
Out Web
Observed Correlations

• Mail traffic up
• MS CPU utilization up
  – Service time up, service load up, service queue longer, latency longer
• DNS CPU utilization up
  – Service time up, request rate up, latency up
• Access edge b/w down

Causality no surprise!

How does mail traffic cause DNS load?
Shape Mail Traffic

Root cause:

- Spam appliance --> DNS lookups to verify sender domains;
- Spam attack hammers internal DNS, degrading other services: NFS, Web
• Shape mail traffic
  – Mail delay acceptable to users?
  – Can’t do this forever unless mail is filtered at the Internet edge

• Load balance DNS services
  – Increase resources faster than incoming mail rate
  – Actually done: dedicated DNS server for Spam appliance

• Other actions?
  – Traffic priority
  – QoS knobs
Analysis

- Root causes difficult to diagnose
  - Transitive and hidden causes
- Key is pervasive observation
  - iBoxes provide the needed infrastructure
  - Observations to identify correlations
  - Perform active experiments to “suggest” causality
Challenges

• Policy specification: how to express? SLOs?
• Experimental plan
  – Distributed vs. centralized development
  – Controlling the experiments … when the network is stressed
  – Sequencing matters, to reveal “hidden” causes
• Active experiments
  – Making things worse before they get better
  – Stability, convergence issues
• Actions
  – Beyond shaping of classified flows, load balancing, server scaling?
Implications: Network Management

- Processing-in-the-Network is real
- Enables pervasive monitoring and actions
- Statistical models to discover correlations and to detect anomalies
- Automated experiments to reveal causality
- Policies drive actions to reduce network stress
Networks Under Stress
Summary

• “DC is the Computer”
  – Prog Sys: RoR, Libraries: Web Services
  – Development Environment: RAMP (simulator), AWE (tester), Web 2.0 apps (benchmarks)
  – Debugging Environment: *Trace + X-Trace

• Near-term Objectives
  – DC Energy Conservation + Reliability Enhancement
  – Web 2.0 Apps in RoR
• Develop-Analyze-Deploy-Operate modern systems at Internet scale
  – Ruby-on-Rails for rapid applications development
  – Declarative datacenter for correct-by-construction system configuration and operation
  – Resource management by System Statistical Machine Learning
  – Virtual Machines and Network Storage for flexible resource allocation
  – Power reduction and reliability enhancement by fast power-down/restart for processing nodes
  – Pervasive monitoring, tracing, simulation, workload generation for runtime analysis/operation