2nd Symposium on Networked Systems Design and Implementation (NSDI '05)

Boston, Massachusetts
May 2–4, 2005

Keynote: The Challenges of Delivering Content and Applications on the Internet

Tom Leighton, MIT/Akamai

Summarized by Ningning Hu

Tom Leighton explained that Internet problems adversely affect current Web services. He pointed out that, for economic reasons, peering links often have limited capacity and that this can easily lead to poor performance, because Internet routing algorithms do not adapt to load. To make matters worse, routing protocols are subject to human errors, filtering, and intentional theft of routes. Tom discussed Internet security issues, working through an example of DNS hijacking. He made the point that virus and worm proliferation and DOS and botnet attacks are severe problems. In 2003, over 10% of PCs on the Internet were infected with viruses. These are not all home PCs: 83% of financial institutions were compromised, double the figure from 2002. Additionally, 17 out of 100 surveyed companies were the target of cyber extortion, and the number of botnet attacks against commercial sites is rising sharply. These problems are very hard to solve, because the Internet was designed around an assumption of trust that is no longer valid.

Tom then described Akamai's on-demand infrastructure. It is made up of around 15,000 servers at 2400 locations on over 1000 networks in 70 countries; Akamai serves 10–15% of all Internet Web traffic each day. On average, Akamai can make small Web sites 15 times faster and large Web sites 2 to 3 times faster. Tom said that studies show that this translates directly into economic gain, e.g., a faster site for a top hotel generates an extra $30 million per year. The core idea of the infrastructure is to choose servers as close as possible to clients so as to avoid Internet problems. This helps because the Internet consists of more than 15,000 networks and none of them controls more than 5% of the total access traffic. Akamai's SureRoute also finds alternative routes via intermediate Akamai servers when the network fails or performs poorly. It monitors roughly 40 alternative routes for each Web site, which improves performance by 30% on average.

Tom finished by highlighting the recent PITAC report on cyber-security, which calls for more investment in fundamental security research.

INTERNET ROUTING

Summarized by Ram Keralapura and Bob Bradley

Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network

Jian Wu and Zhuoqing Morley Mao, University of Michigan; Jennifer Rexford, Princeton University; Jia Wang, AT&T Labs—Research

Morley Mao described a tool that monitors BGP (Internet routing) updates to find in real time a small number of high-level disruptions (such as flapping prefixes, protocol oscillations due to Multi-Exit Discriminators, and unstable BGP sessions). Unlike earlier research, it does not focus on finding the root cause of routing changes. The problem addressed is important because route changes are common and are associated with congestion and service disruptions; the hope is that operators can use notifications from the new tool to further mitigate the situation for users. It is challenging because there are many possible reasons for a given routing update, and multiple updates can originate from one underlying event, and it is difficult to decide which events are significant to operators.

The tool works by capturing BGP updates from border routers that peer with larger networks. This data is fed into a centralized system which processes the updates in real time. It groups the updates, classifies them into events, correlates the events, and then predicts traffic impact. A key difficulty is the large volume of BGP updates (there are millions daily). The discussion raised the issue of looking at data traffic directly, since significant events are by definition those that affect data traffic.

Design and Implementation of a Routing Control Platform

Matthew Caesar, University of California, Berkeley; Donald Caldwell, Aman Shaikh, and Jacobus van der Merwe, AT&T Labs—Research; Nick Feamster, MIT; Jennifer Rexford, Princeton University

The motivation for the authors was basic design issues in the iBGP protocol connecting routers within ISPs. Current full-mesh iBGP doesn't scale, is prone to protocol oscillations and persistent loops when used with route reflection, and is hard to manage and difficult to develop. Their RCP approach attempts to address each of these problems by computing routes from a central point and removing the decisions from the routers. Use of a centralized system brings up the problem of single point of failure. The authors address this issue by replicating RCP at strategic network locations. They argue that, unlike route reflection, there will be no consistency issues that could potentially result in problems like forwarding loops. Matt argued that the RCP system has better scalability, reduces load on routers, and is easier to manage because it is configurable from a single point. It is also deployable, because it does not require changes to closed legacy router software. While RCP is only a first step at this stage, these properties may make it a practical way to improve Internet routing.

- ***Negotiation-Based Routing Between Neighboring ISPs***

Ratul Mahajan, David Wetherall, and Thomas Anderson, University of Washington

Today's Internet is both a competitive and a cooperative environment, because ISPs are self-interested but carry traffic for each other. Each ISP independently decides how to route its traffic and optimizes for different points, and ISPs don't share internal information. This can result in inefficient paths and unstable routes. Tom Anderson presented a negotiation model the authors developed to help solve these problems. It tries to find a point between cooperation and competition that limits the inefficiencies. ISPs assign preferences for routing options using an opaque range. They then exchange these preferences and take turns picking better routing options. They can reassign preferences when needed, and the process stops when either ISP wants it to. This strategy respects ISPs' self-interest by allowing them to barter route choices according to their preferences (with each ISP losing a little on some flows and gaining more on others). ISPs have incentives to find good compromises because each stands to win overall and has no risk of losing. The goal is for both fair play and overall win-win results. The scheme was evaluated by simulation, which found that ISPs can achieve close to the socially optimal routing even though they must both win. Future work includes multiple-ISP negotiations.

The question of cheating came up in the discussion. The authors explored simple cheating strategies and argued that there is little incentive to cheat, as the cheater often does less well than if he hadn't cheated. Another point of discussion was how well the scheme would work for traffic engineering, where preferences change depending on the load.

MODELS AND FAULTS

Summarized by Matthew Caesar

- ***Detecting BGP Configuration Faults with Static Analysis***

Nick Feamster and Hari Balakrishnan, MIT

Awarded Best Paper

Nick Feamster presented RCC, a router configuration checker that uses static analysis to detect faults in BGP configurations. Today, checking is highly ad hoc. Large configuration faults do occur and can cause major outages. Nick gave a taxonomy of faults. The goal of the RCC is to allow configurations to be systematically verified for correctness before being deployed. Correctness is defined in terms of two goals: path visibility (if there's a path between two points, the protocol should propagate information about the path) and route validity (if there's a route, there exists a path). RCC uses goals to produce a list of constraints and checks these constraints against the configurations. It was evaluated against configurations from 17 different ASes. It succeeded in uncovering faults without a high-level specification of the protocol. The major causes of errors were distributed configuration and the complexity of intra-AS dissemination (as configuration often expresses mechanism, not just policy). RCC is available online.

Q: Do large, well-run ISPs generate router instance configurations in a centralized manner? Would RCC provide any benefit in this case?

A: Many ISPs run scripts from a centralized database, but many do not, and even with a centralized database there can be errors (e.g., bad copy/pastes).

Q: What is the number of constraints you solved for most networks?

A: We used a fixed set of constraints resulting in a polynomial time algorithm.

- ***IP Fault Localization via Risk Modeling***

Ramana Rao Kompella and Alex C. Snoeren, University of California, San Diego; Jennifer Yates and Albert Greenberg, AT&T Labs—Research

Ramana Kompella presented SCORE, a tool that identifies the likely root causes of network faults, especially when they occur at multiple layers. Today, troubleshooting is ad hoc, with operators manually localizing faults reported via SNMP traps. This is challenging because alarms tell little about the failure; network databases can be corrupt or out-of-date, networks are highly layered (35% of the links have >10 components), and correlated failures can occur (e.g., a single fiber cut can take down several links).

SCORE constructs a Shared Risk Link Group (SRLG) database that provides a mapping from each component to a set of links that will fail if the component fails. It manipulates this as a graph, using greedy approximations to find the simplest hypothesis to explain failure observations. SCORE also allows for imperfections (e.g., lost observations) with an error threshold. It performed well in practice: The accuracy was 95% for 20 failures; the misdiagnoses were due to loss of failure notifications and database inconsistencies. Ramana mentioned probabilistic modeling of faults and other domains (MPLS, and soft faults like link congestion) as future work.

Q: Would it be practical to use steady-state conditions to improve your results, e.g., if you assume the network is working correctly most of the time?

A: You could inject faults into a network and test, but most ISPs wouldn't be willing to do that.

Q: You were able to uncover inconsistencies in the database. But isn't this circular: How do you know your inferences were correct if they come from an incorrect database?

A: You have to assume the database is reasonably accurate. Unfortunately, you can't just query the system to find out the IP/optical relations.

- ***Performance Modeling and System Management for Multi-Component Online Services***

Christopher Stewart and Kai Shen, University of Rochester

Online services that run on clusters in heterogeneous environments are difficult to model, predict, and manage. There is work on performance models to guide provisioning for single-component services, but it is not adequate when multiple components in the system can be replicated and interact with each other in complex ways. Christopher Stewart described a profile-driven approach to model system performance. It works at the OS level to profile key application characteristics transparently. They predict the resources required for individual components and transparently capture communications at the system-call level to model interconnections. Different models are then constructed for throughput and response time. The authors compared the resulting predictions with the actual system performance and found them to be accurate within 1%.

Q: Does it make sense to try real-time feedback to improve the model online?

A: Yes. We did it offline, but one could refine our approach in an online fashion.

Q: Have you considered bottlenecks in real machines' CPU/memory/network?

A: Yes, we do this in our model.

Q: What kinds of application behaviors would make your accuracy poor? For example, would caching effects reduce your accuracy?

A: We address caching and some other issues in the paper, but there could be interesting future work in that direction.

OVERLAYS AND DHTS

Summarized by Bernard Wong

- ***Debunking Some Myths About Structured and Unstructured Overlays***

Miguel Castro, Manuel Costa, and Antony Rowstron, Microsoft Research Cambridge

Popular file-sharing applications such as Gnutella use unstructured overlays that do not constrain links between nodes and rely on flooding to spread queries. To improve scalability, structured overlays constrain node and link placement so that queries can be resolved in O(log n) hops. However, people have claimed that structured overlays are unsuited for real-world applications given churn, heterogeneity, and complex queries. Miguel Castro's talk focused on debunking these myths using a trace-based simulation of Pastry and Gnutella 0.4. He showed how methods to handle heterogeneity that mimic unstructured techniques can be added to structured overlays. Similarly, he described structured flooding and random walks for complex queries.

Q: One advantage of unstructured overlays is that the overlay structure is decoupled from the service structure, allowing for reuse between services. Can you comment on this?

A: We could reuse structured overlays too. For example, we can carve out a part of a larger structured overlay for a single smaller service.

Q: How do heartbearts scale?

A: Heartbeats are sent at a fixed rate, independent of system size. The total overhead is fairly low.

Q: What are your thoughts on Mercury?

A: Mercury is a hybrid network with constrained routing that can solve complex queries. It emulates the functionality of unstructured overlay, but cannot solve arbitrary queries, such as matching based on regular expressions.

- ***Bandwidth-Efficient Management of DHT Routing Tables***

Jinyang Li, Jeremy Stribling, Robert Morris, and M. Frans Kaashoek, MIT

Accordion is a DHT (distributed hash table) that addresses the trade-off between maintenance overhead and lookup performance. Reduced maintenance traffic leads to lower lookup performance due to churn, while aggressively maintaining neighbor freshness can be expensive in terms of bandwidth. Choices for maintenance frequency are often uninformed, since a priori knowledge of the churn rate is not usually available. Instead, Accordion relies on an outbound bandwidth budget to limit the amount of maintenance. It discovers new nodes, tracks the probability of a neighbor being dead (based on the lifetime of the neighbor and time of its last communication), and removes those whose probability exceeds a fixed threshold. Compared with Chord and OneHop, Accordion achieves lower average lookup latencies for a given average bytes per node per second alive.

Q: Small-world properties are based on a neighbor distribution that is the inverse of the distance in the ID space. Would opportunistic neighbor discovery change the distribution and the properties?

A: If lookup keys are not uniform, then it is not guaranteed to yield small-world characteristics.

Q: What if nodes choose to behave maliciously in order to meet bandwidth budget?

A: Accordion is not designed to work in a malicious environment.

- ***Improving Web Availability for Clients with MONET***

David G. Andersen, Carnegie Mellon University; Hari Balakrishnan, M. Frans Kaashoek, and Rohit N. Rao, MIT

The end-to-end availability of the Internet (95% and 99.6% in earlier studies) compares poorly to standard phone service. MONET aims to achieve 99.9% to 99.99% availability by exploiting the path and replica diversity that exists for Web downloads. It consists of an overlay of squid Web proxies and a parallel DNS resolver. The key difficulty is that the number of paths through the overlay to all replicas can be large. This is good for diversity, but bad for overhead if all paths are to be explored. In MONET, a waypoint selection algorithm returns a set of paths separated by delays. These paths are most likely to be successful, based on previous path history, and are explored in order to minimize overhead.

A six-site MONET has been deployed for two years with approximately 50 users per week. Its waypoint algorithm achieves availability that is similar to using all possible paths. This is 99.99%, if server failures are discounted. Also, Akamai sites have eight times more availability than non-Akamai sites if server failures are included in the availability metric. A challenge in gathering real measurements was the many incorrectly configured DNS and Web servers; consistently unreachable services were discounted in the measurements.

Q: When performing parallel connections, does MONET just perform the TCP connect, or does it download the object twice?

A: MONET just performs the TCP connect.

Q: Would MONET choose lossy but low-latency links?

A: Previous studies have shown that the first SYN packet is a good predictor of how long it will take to download the desired content over the connection.

STORAGE

Summarized by Kevin Walsh

- ***Shark: Scaling File Servers via Cooperative Caching***

Siddhartha Annapureddy, Michael J. Freedman, and David Mazières, New York University

Siddhartha Annapureddy presented Shark, a file system that is as convenient and familiar as NFS, yet scales to hundreds of clients and supports cross-file-system sharing. Pushing bundles of software to the nodes of a distributed system is wasteful, even with dissemination systems such as BitTorrent, because not all of the software may be needed. Instead, what is needed is the illusion that all files are located on every node, with the files being fetched only as needed. NFS provides these semantics but does not scale well, because a large number of clients cause delays at the central server. On the other hand, P2P file systems do scale, but have nonstandard administrative models and new semantics, and so are not widely deployed. Shark combines both advantages by using a central server model together with very large cooperative client caches (to reduce redundant traffic at the server). One intriguing idea was to allow chunks of data to be shared across file systems, increasing the effective size of the cache. Several security concerns were discussed: clients need to be able to check integrity, eavesdroppers should not be able to see the contents, and cache sharing is somewhat in conflict with privacy. In one PlanetLab test, Shark retrieved a 40MB package in seven minutes, compared to 35 minutes for SFS. Another test revealed an eightfold improvement over NFS in the number of bits pushed through the network.

Q: How would a least-common-chunk fetch ordering policy, like BitTorrent, compare with your sequential or random orderings?

A: We could do that, but we did not look at it yet.

Q: What consistency guarantees do you provide while things are being transferred?

A: We guarantee NFS-style consistency semantics at all times. This is done with leases at the central server.

Q: Your chunk cache indexes data only by the hash of the chunk. What do you do in case of hash collisions?

A: We assume there will be no hash collisions. This is the standard assumption for these scenarios.

Q: You showed scalability of Shark in terms of bandwidth, but the server is involved in each chunk transfer, no?

A: The authentication and session keys are between client and client, not client and server. The client must initially talk to the server to get chunk tokens, but then goes to clients to get chunks. This potentially uses many RPCs if the file is very large.

- ***Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures***

Andreas Haeberlen, Alan Mislove, and Peter Druschel, Rice University

A common assumption in distributed storage systems is that diversity is high because nodes use different OSes, applications, administrators, users, etc. This results in independent failure models, so that reliability comes from a small amount of replication. But these are unrealistic assumptions in practice: 70–80% of the OSes in use are Windows, and a virus or worm can lead to a correlated failures that spreads too rapidly even for reactive approaches to respond. So what can we do? Glacier's approach is to use massive redundancy to tolerate correlated failure rather than try to predict and exploit correlations (like Phoenix and OceanStore). Of course, there is an upper bound on the maximum number of failures, but it is easier to pick this number than to specify a complete failure model.

The central question is whether this can be done with a reasonable amount of storage and bandwidth. Glacier uses erasure codes with a high degree (50 or 100 fragments per object, with only about 5 needed to recreate the object) and replicates data, too. Even during a correlated failure there should be enough fragments to reconstruct objects. A risk in Glacier is that objects may expire during a correlated failure. Also, the per-file overhead is especially large for small files.

Glacier was evaluated with a trace-driven workload and deployment with 17 users and 20 nodes based on FreePastry, PAST, Scribe, and Post. An artificial 58% correlated failure induced no losses of data at all. Glacier has yet to see any loss of data in deployment.

Q: In your test system, you use 5/48 encoding even though you had only 20 nodes. Couldn't you just use 5/20 nodes?

A: We wanted an idea of a realistic overhead. During the experiment, the size of the system grew and changed, and we felt 5/48 would be more realistic for a larger system.

Q: If you knew the size of the system, would you set the number of fragments to equal the number of nodes?

A: Normally there would be many more nodes than fragments.

Q: Won't the downtime constant cause poor performance because it is fixed and will be a poor choice sometimes?

A: The one-week figure came from an assumption that users could not do without email for more than a week. In reality, users were using more than one email system and sometimes let their node remain offline for more than a week. We have switched to four weeks, but perhaps could do something automatic.

Q: How do you assign fragments to nodes, and how do nodes know which fragments to store?

A: The assignment of fragments to nodes is done by the hash of the fragment. We divide the ring into 48 sections, and store at hash+1x, hash+2x, ..., hash+48x.

Q: How did you know not to store a fragment on the node that was down at the time of insertion?

A: The neighbors in the ring keep the pointer to the down node for one week, and can then report the node as being down whenever a message is destined for that node.

BUILDING NETWORK SERVICES

Summarized by Ashwin Sampath

■ *Quorum: Flexible Quality of Service for Internet Services*

Josep M. Blanquer, Antoni Batchelli, Klaus Schauser, and Rich Wolski, University of California, Santa Barbara

Internet services such as e-commerce tend to be clustered architectures in which it is important to provide acceptable levels of service to different kinds of customers. Current solutions either throw hardware at the problem (overprovisioning) or embed QoS logic in the application code. This is expensive either in terms of equipment or in reprogramming time. Josep Blanquer presented Quorum, which tackles these problems while being readily deployable. Quorum provides its QoS guarantees at the boundaries of an Internet site. This is an effective location to classify user requests into service classes and shape traffic based on the priorities of incoming requests, without delving inside the cluster. The authors show this by evaluating their solution on a 68-CPU cluster with the Teoma Internet search service alongside five other QoS architectures. They also examined the effects of sudden fluctuations in traffic and cluster node failures.

■ *Trickles: A Stateless Network Stack for Improved Scalability, Resilience, and Flexibility*

Alan Shieh, Andrew C. Myers, and Emin Gün Sirer, Cornell University

Today's client-server applications are built on TCP/IP, which stores per-connection state at both ends. This limits scalability (due to memory constraints) and leaves the server vulnerable to denial of service. Alan Shieh presented Trickles, a radical alternative in which the server state is moved to the clients. Each client supplies transport and user continuations along with their packets to request any computation. The server establishes a context based on these continuations, performs the requested computation, updates the associated state, and sends it back to the client along with the result of the computation. To make this work, the authors implemented an event-based server API. This design lends itself to efficient server load balancing schemes and transparent server failover mechanisms, because clients establish contexts before issuing each computation request. A typical target application is a busy Web server. Alan presented an evaluation that showed the memory overhead of Trickles to be lower than TCP/IP and the throughput rates to be comparable.

■ *Designing Extensible IP Router Software*

Mark Handley, University College, London/ICSI; Eddie Kohler, University of California, Los Angeles/ICSI; Atanu Ghosh, Orion Hodson, and Pavlin Radoslavov, ICSI

Everyone wants to fix BGP in some way (convergence, security, scalability), but the size of the routing infrastructure and expectations of 99.999% uptime make experiments with routing software almost impossible. Mark Handley presented XORP, IP routing software designed for extensibility, latency, and scalability. XORP is based on an event-driven architecture with emphasis on quick processing and propagation of routing changes between processes. This lends itself to extensibility and experimentation since each process is independent. XORP's BGP implementation is based on a data flow model, with routing tables implemented as processes that pass along routing updates. This differs from conventional router software designs, where all routing protocols process routing updates and store routes in a single large table. The trade-off is that the modular and robust design of XORP marginally increases memory usage but results in faster routing convergence. To show this, the authors tested the convergence times of Cisco(IOS), Quagga, and MRTD: Cisco and Quagga routers take up to 30 seconds to converge, while MRTD and XORP are consistently under one second.

WIRELESS

Summarized by Ashwin Bharambe

■ *Using Emulation to Understand and Improve Wireless Networks and Applications*

Glenn Judd and Peter Steenkiste, Carnegie Mellon University

Most wireless network studies are performed in simulation, which can be carefully controlled but misses many realistic factors. Glenn Judd proposed an emulation infrastructure to bridge the gap between simulation and real testbed evaluation. The basic idea is to use real wireless NICs at the sender and receiver and to control signal propagation through a customized FPGA. Analog signals from the sender are down-sampled and converted to digital format, processed by a DSP engine (built using the FPGA), converted back to analog format, and fed to the wireless antenna at the receiver. Glenn presented results validating the hardware. He also showed that different wireless cards from the same manufacturer and card family have surprisingly different RSSI and noise characteristics. In the discussion, it was suggested that Glenn compare the results of using the emulator with that of simulation models in simulators like QualNet and ns-2.

■ *Geographic Routing Made Practical*

Young-Jin Kim and Ramesh Govindan, University of Southern California; Brad Karp, Intel Research/Carnegie Mellon University; Scott Shenker, University of California, Berkeley/ICSI

Young-Jin Kim described the Cross-Link Detection Protocol (CLDP) for enabling geographic routing. Previous geographic routing (GPSR, Greedy Perimeter Stateless Routing) is based on face traversal with the right-hand rule. This needs a perfect planarization of the radio graph to operate correctly, and fails in practice due to irregular localization of wireless cards and radio-opaque obstacles. The previously proposed "mutual witness" fix also suffers from problems: It generates some additional cross-links and can result in collinear links as well. CLDP discovers and removes cross-links in a radio graph. It leaves some cross-links to prevent network partitions, but guarantees that face traversal will never fail. CLDP was evaluated using the TinyOS simulator with 200 nodes and 200 obstacles. It outperformed previous geographic routing protocols in terms of maintaining reachability and providing low stretch.

Q: Does CLDP work under dynamic conditions?

A: Yes, if the velocity of the nodes is limited.

■ *Sustaining Cooperation in Multi-Hop Wireless Networks*

Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan, University of Washington

Maya Rodrig presented Catch, an add-on to multi-hop wireless routing protocols to deter "free-riding," in which nodes use the network but decline to forward packets. The protocol first detects free-riding behavior, then leverages the majority of "good" nodes to punish the "bad" node. The key idea was to send anonymous probes to which neighbors must respond. This forces a potentially bad node in the network to reveal its connectivity to everybody. Furthermore, packets relayed by a node can be overheard, due to the broadcast nature of the medium. Detection thus boils down to checking whether more data packets (which were meant to be forwarded) are dropped as compared to the anonymous probe responses. The protocol also incorporates a strategy based on one-way hash functions to enable neighbors to punish a misbehaving node. Handling attacks based on signal strengths is future work.

Q: What about Sybil attacks?

A: Catch builds on unforgeable identities for nodes.

Q: Can you falsely accuse a "good" node?

A: Yes, in which case Tit-for-Tat retaliates.

SYSTEM MANAGEMENT AND CONFIGURATION

Summarized by Sherif Khattab and Dushyant Bansal

- ***ACMS: The Akamai Configuration Management System***

Alex Sherman, Akamai Technologies and Columbia University; Philip A. Lisiecki and Andy Berkheimer, Akamai Technologies; Joel Wein, Akamai Technologies and Polytechnic University

Akamai's CDN (Content Delivery Network) serves Web content using 15,000+ edge servers deployed in 1,200+ ISPs. Its configuration information comes from Akamai customers, who want to control how their content is being served via hundreds of parameters (e.g., cache TTL, allow lists, cookie management) and internal Akamai services such as mapping and load balancing. Alex Sherman presented the ACMS system for timely, reliable delivery of dynamic configuration files in this system. ACMS is composed of front ends that accept, store, and synchronize configuration file submissions, and back ends that deliver configuration files to edge servers. It uses a quorum-based protocol for agreement and synchronization among the front ends. Recovery is optimized using snapshots, a hierarchical versioning structure. Edge servers download configuration files via Akamai's CDN with hierarchical caching. ACMS is divided into zones that are tested incrementally to avoid systemwide effects from bad configuration files. During the first nine months of 2004, 36 network failures affected the front ends, and in over six months of 2004 there were three recorded instances of file corruption. ACMS continued to work successfully. It took about two minutes to submit and deliver most configuration files. An audience member asked about TTL versus cache invalidations. Sherman responded that the TTL technique is easier and tolerates propagation delays. However, for some cases, Akamai uses cache invalidation.

- ***The Collective: A Cache-Based System Management Architecture***

Ramesh Chandra, Nickolai Zeldovich, Constantine Sapuntzakis, and Monica S. Lam, Stanford University

About 30,000 desktops are infected every day, and downtime and confidentiality breaches translate into monetary damage. Ramesh Chandra presented the Collective, a cache-based system to improve the management of desktop PCs. It trades customizability for manageability through centralized management and distributed computation. The Collective introduces the concept of a virtual appliance, an encapsulation of system state (OS, shared libraries, and installed applications). Examples include Windows XP, GNU/Linux with NFS, and GNU/Linux with local disk. Appliances are stored in appliance repositories editable only by administrators, whereas user state (user preferences and data) is stored in data repositories. In the Collective, software updates are atomic and dependable. Caching provides support for disconnected operation, a useful feature for mobile users: Chandra described a USB memory stick carrying appliances and data. A prototype of the Collective has been used for about a year on a daily basis at Stanford. Users find the system to be simple, with low virtualization overhead. From a 15-day block read trace, 80% of requests were for 20% of the data. Answering a question from the audience, Chandra identified graphics applications and 3-D games as unsuitable for usage in the Collective.

- ***Live Migration of Virtual Machines***

Christopher Clark, Keir Fraser, and Steven Hand, University of Cambridge Computer Laboratory; Jacob Gorm Hansen and Eric Jul, University of Copenhagen; Christian Limpach, Ian Pratt, and Andrew Warfield, University of Cambridge

It takes about eight seconds to move the memory of a Virtual Machine (VM) over a machine cluster running Xen with networked storage, good connectivity, and support for L2 or L3 traffic redirection. Meanwhile, live interactive applications, such as Web servers, game servers, and quorum protocols, have soft real-time requirements. Ian Pratt presented a technique for relocating interactive VMs with downtime as low as 60ms. It uses iterative, rate-limited pre-copy of VM memory while the VM continues to run. Pre-copy is more effective than on-demand page faulting and leaves no "residual dependencies" on the original host. Pratt introduced the concept of the Writable Working Set (WWS) of a VM. They represent hot pages, such as process stacks, and network receive buffers. The size and dirtying rate of WWS are crucial in determining the number and rate of pre-copy iterations. Pratt also presented results for relocating a Web server running the SPECWeb benchmark, a Quake3 game server, and a synthetic worst case with rapid page dirtying.

SECURITY

Summarized by Robert Picci

- ***Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds***

Srikanth Kandula and Dina Katabi, MIT; Matthias Jacob, Princeton University; Arthur Berger, MIT/Akamai

Awarded Best Student Paper

Srikanth Kandula focused on CyberSlam attacks, in which an attacker harnesses potentially hundreds of thousands of "bots" spread across the Internet to take down a Web site. The key feature of these attacks is that they attempt to exhaust resources on the server by making requests that are indistinguishable from those of legitimate clients. Srikanth presented a novel defense based on CAPTCHAs, the graphical reverse Turing tests used to prevent automated account signup. When a CyberSlam attack or flash crowd is detected, the system starts using CAPTCHAs to distinguish legitimate users from attackers. They are served without per-client state at the server. Once it has learned which clients are the attackers (they cannot solve CAPTCHAs), the system switches into a mode where known attackers are kept out and new users are allowed in without CAPTCHA tests. Admission control is also used to balance system resources between authenticating new users and serving those who have proven themselves legitimate. This improves server responsiveness, not only under attack, but also under flash crowds.

- ***Cashmere: Resilient Anonymous Routing***

Li Zhuang and Feng Zhou, University of California, Berkeley; Ben Y. Zhao, University of California, Santa Barbara; Antony Rowstron, Microsoft Research, UK

Cashmere addresses some weaknesses in existing anonymous routing by using a structured overlay (in this case, FreePastry). Li Zhuang began with the basic idea of secure anonymous routing: Packets are sent to their destinations through a series of intermediaries such that no one but the sender knows the entire path; cryptography is used to hide routing information from nodes as well as to protect the message contents. Without massive collusion, no one knows who sent the packet, and only the receiver can see its contents. However, with earlier schemes, failed intermediaries can reduce reliability, and the cryptography can be expensive. Cashmere deals with failures by exploiting the overlay to route each packet to a group of nodes rather than a single node. This makes it more likely for packets to get through when there is churn. Cashmere reduces the amount of per packet cryptographic computation by decoupling the payload from the routing information. Session keys and lightweight symmetric ciphers can then be used instead of public-key cryptography.

SENSOR NETWORKS

Summarized by Rebecca Braynard

- ***Decentralized, Adaptive Resource Allocation for Sensor Networks***

Geoffrey Mainland, David Parkes, and Matt Welsh, Harvard University

Matt Welsh talked about controlling sensor network resources in a distributed manner by using market prices. Nodes determine their actions using a globally known reward: local available energy and data dependencies. These actions include listening for incoming messages and taking sensor readings. The algorithm is motivated by the example of tracking a tank in a field of sensors and is evaluated through a 100-node simulation with the metrics of accuracy, energy consumption, and energy efficiency. The mechanism uses less energy to track an object and is more effective at adapting to changing conditions. The authors plan to develop richer models that extend allocation across multiple users and queries and adjust reward settings during runs.

Q: Can the pattern of movement lead to dead nodes?

A: The energy budget of a node limits its consumption.

Q: Since nodes have a local view, can they get caught in a "busybody" situation?

A: Yes, nodes can get caught in loops, and feedback is needed.

Q: With a TinyOS model you can meet resource allocation guarantees. You can't with your approach. Which is better?

A: Periodic duty cycling is good for some applications, but not all.

- ***Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensornets***

Rodrigo Fonseca, Cheng Tien Ee, David Culler, and Ion Stoica, University of California, Berkeley; Sylvia Ratnasamy, Intel Research; Jerry Zhao, ICSI; Scott Shenker, University of California, Berkeley/ICSI

Rodrigo Fonseca presented BVR, a simple routing protocol that only uses local state and does not depend upon geographic locations. Instead, BVR creates a virtual coordinate space with connectivity information. In the algorithm, r nodes are chosen to be beacons, and the remaining nodes find their distances to the beacons. To transmit a packet, a node uses the destination location to route packets through the neighbor closest to the destination (greedy algorithm). If the nodes are in a local minimum, they send the packet through the closest beacon node. If the greedy algorithm does not work, the packet is flooded through the network. BVR was evaluated with a high-level simulation (3200 nodes), an implementation on Mica2 Motes, and a low-level simulator, TOSSIM. It was found to outperform a greedy geographic routing protocol.

Q: Will the beacons be running out of power prematurely?

A: Not necessarily, since the data does not go through the beacons, so they may not consume more power.

- ***Active Sensor Networks***

Philip Levis and David Culler, University of California, Berkeley; David Gay, Intel Research

Phil Levis argued that sensor networks cannot realize their potential given the energy consumption associated with existing frameworks. Sensor networks often need to be reprogrammed after deployment, as it's not efficient to collect all data and process it offline. Yet they do not need to be reprogrammed, since the networks are application-specific. Instead, an application-specific virtual machine (ASVM) can be used. This leverages the trade-off that many cycles can be performed by a sensor node for each bit sent or received. ASVMs provide a flexible, simple, and efficient infrastructure for programming devices. (See the paper for details on their design.) To show their effectiveness, Phil compared the original and the VM implementations of a region library (Regions Fiber) and query library (TinyDB/TinySQL) on a 42-node testbed.

Q: To provide concurrency, the Banker's algorithm is used; does this create a disadvantage for allocating resources?

A: It is a conservative approach and a drawback. To reduce the impact, programmers should use short-running handlers.

Q: In many projects, the work is to overcome small amounts of memory. Given Moore's Law, should this work be focused on energy consumption instead of memory management?

A: Memory is limited by energy; this will affect how much memory is available and how it is used.