# First-Class Access For Developing-World Environments


Vivek S. Pai
Department of Computer Science
Princeton University
35 Olden Street
Princeton, NJ, USA
vivek@cs.princeton.edu

Anirudh Badam
Department of Computer Science
Princeton University
35 Olden Street
Princeton, NJ, USA
abadam@cs.princeton.edu

Sunghwan Ihm
Department of Computer Science
Princeton University
35 Olden Street
Princeton, NJ, USA
sihm@cs.princeton.edu

KyoungSoo Park
Department of Electrical Engineering
KAIST
335 Gwahangno, Yuseong-gu
Daejeon, Republic of Korea
kyoungsoo@ee.kaist.ac.kr



## ABSTRACT

Improvements in connectivity and the cost of laptops will soon enable widely-available Internet access in parts of the world where access ranged from rare to unavailable. While such steps represent tremendous progress, we believe that the next barrier to adoption will be to convince providers that the gap between "good enough" access and first-class access, while quite large, is possible to bridge. This paper presents our views for the creation of a networking software stack tailored toward developing-world usage. With this stack, we believe that developing-world users will have a participatory Internet experience, similar to that of users in the US and Europe. The stack is also geared toward a low resource footprint, making it deployable in cost-sensitive environments. We have completed some portions of the stack, and believe that the total system represents a three to five year effort.



## Categories and Subject Descriptors

C.2.2 [**Network Architecture and Design**]: Applications; C.2.4 [**Distributed Systems**]: Client/server

## General Terms

Design, Experimentation, Performance

## Keywords

Web Caching, WAN Acceleration, Developing Region




*CFI* '10 June 16–18, 2010, Seoul, Korea.



## 1. INTRODUCTION

While the gap between the "haves" and "have-nots" in Internet access is wide, the gap between the "haves" and the "almost-haves" may not be much better. As the first world moves toward user-generated content, social networking, blogs, comment-driven sites, and more participation, having anything less than full access to the Internet will degrade the Internet user experience. While many people believe that read-only offline access to the Internet is "good enough", we believe that this approach will hinder developing-world users from sharing information not only with the developed world, but also with each other, which appears to be a largely unaddressed desire [21]. In the longer term, we also believe that offline-only access will fail to spur the kind of Internet growth seen in the first world.

While being a second-class Internet citizen is no doubt better than being excluded completely, a number of technological advances may soon render the choice between first-class and second-class a false dichotomy. Low-cost laptops can bring personal computing to large numbers of people [11, 13]. Long-range wireless can bring connectivity where no connectivity existed [16]. Large-capacity low-cost disks can provide bulk storage that transforms how developers think of data retention. Solid-state disks can boost the performance of out-of-core applications. Current low-cost laptops combined with USB-attached hard drives can provide this level of hardware for $400 USD per unit, and this cost may drop over time.

Our focus, at a high level, is to use these technologies to narrow the gap between the usage experiences of the developed world and the developing world. This combination will likely mean that our focus will be on the growing urban middle class and the upper-middle class in particular, but even this target audience is sizable – if we assume that one-quarter of India's and China's population falls into this category, the number of users exceeds

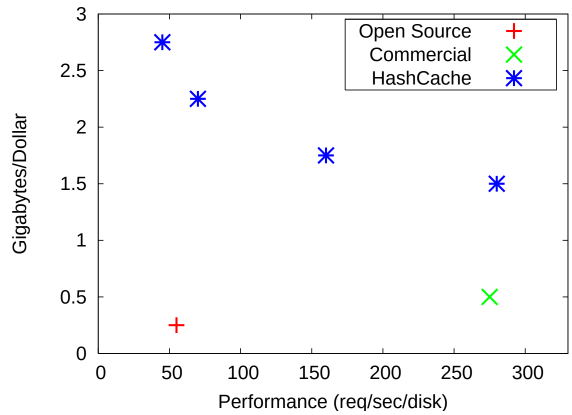


**Figure 1: Hardware cost/performance for HashCache variants and other proxies. These figures include the cost of the machine and memory needed to achieve the given performance level. The four HashCache values represent different configurations.**

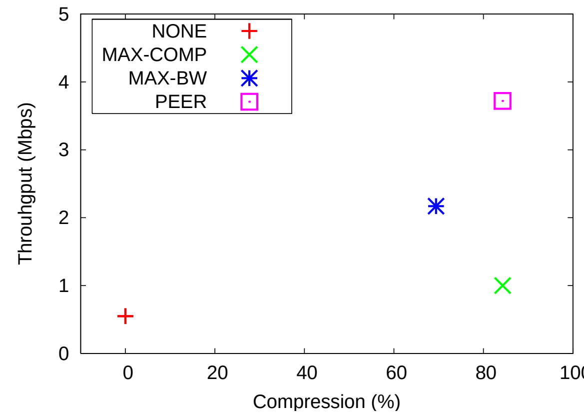


**Figure 2: Compression/performance tradeoffs for our WAN accelerator. With only one machine and a single disk, the system is disk-bound at its highest compression rate. When higher throughput is needed, the compression rate can be decreased, or other machines can be used as peers to increase the aggregate disk performance.**

the total population of the United States. We target this audience based on the observation that even if $400 USD is a large value in local currency, many middle-class parents in these countries view it as an investment in their children's education. If such technology can provide a usage experience similar to that of the developed world, it also provides self-empowerment rather than charity, with a family laptop seen as an aspirational item, akin to a television, scooter, or car.

We also view targeting the urban middle class as a means of helping build local ecosystems. Online access can more easily drive advertising and advertising-based purchases, both of which subsidize the cost of developing and delivering content. As more people use the online Internet, the fixed costs of traffic delivery are spread across more users, lowering the cost of delivery, which can then generate more demand from more users. Commercialization of the Internet in the US has generated so much volume that the researchers who originally used the Internet can now buy bandwidth and access much more cheaply than prior to commercialization. We hope that lowering the cost of online access in the developing world can generate a similar effect.

## 2. A DEVELOPING WORLD STACK

To achieve these goals in a way that best exploits our backgrounds, we intend to focus our efforts on a networking software stack tailored toward developing-world usage. The main goals of this stack will be focused on improving the perceived bandwidth and latency of Web applications by localizing activity as much as possible, and moving activity to where it can be most efficiently served. The components of our network stack include: a static Web cache, a WAN accelerator, bandwidth shifting, prefetching, snooping, off-line access, and local search.

### 2.1 Caching

At the heart of our networking stack is a caching storage system called HashCache [3], which enables terabyte-sized caches to be shared among applications, while providing selectable trade-offs between RAM consumption and performance. In its lowest-performance mode, HashCache requires no main-memory indexing. From a developer's standpoint, HashCache coupled with large disks provides a practically-infinite cache store at low cost. Assuming connectivity of one megabit per second, a one-terabyte disk is sufficient to store all communication for three months. A low-overhead cache system combined with this kind of storage capacity frees the developer from having to wonder if some data should be stored, or whether prefetching will pollute the cache – if the least-recently used data on disk is three months old, eviction is not an issue. The HashCache Web Proxy provides a standards-compliant cache for static content.

A comparison of HashCache configurations versus other proxies is shown in Figure 1 on the Polygraph benchmark [8]. In its lowest-performance mode, HashCache performs comparably to Squid (an open source proxy [20]) while using a small fraction of the resources. In practical terms, HashCache can run on a netbook while comparable systems need dedicated server-class machines. In its highest-performance mode, HashCache compares favorably with high-end commercial systems, though the cost-performance difference narrows.

### 2.2 WAN Acceleration

On top of HashCache, we layer Wanax, a WAN accelerator (network packet cache) designed for disks with large capacities but low seek rates [10]. By using HashCache's indexing, it can operate with a very low memory footprint, and the two systems can comfortably share the hardware of a low-end laptop (256MB RAM) with a USB-attached 1TB drive. Commercial WAN accelerators often advertise the fact that they do not store redundant content as a feature, since all content must be indexed (presumably in RAM). In contrast, with HashCache eliminating the RAM pressure from indexing, storing data redundantly on disk can reduce the number of seeks needed. Wanax also provides a peering protocol, allowing content to be fetched from closer peers when possible.

The benefits of this design are shown in Figure 2, where the bandwidth and compression rate are shown for three different configurations. The first (`MAX-COMP`) is the maximum compression configuration, which requires the most disk accesses. The second (`MAX-BW`) enables intelligent load shedding, which can trade a lower compression rate for a higher bandwidth when the wide-area link is available. The third configuration (`PEER`) is with multiple peers in the same region, where high compression and high bandwidth are achieved.

For higher-performance environments, HashCache can use a different indexing scheme, which requires a larger RAM footprint. We intend to use SSDs in these environments, since current low-cost laptops use SSDs that offer performance and capacity between that of their RAM and external hard drives. For these environments, a low-end laptop (256MB RAM) using all of its SSD (typically 4GB) for indexing can provide performance comparable to larger servers.

## 2.3 Bandwidth Shifting

Bandwidth-shifting is the term we use to describe moving bandwidth consumption from a high-cost location to a low-cost location. While Web caches are single-sided (i.e., the client knows when content is cacheable and can avoid contacting the server), WAN accelerators operate in pairs, with one end fetching the content from the server. Bandwidth in the developing world is often more expensive, even in absolute terms, than in high-bandwidth countries. We intend to exploit this difference by providing WAN acceleration endpoints in lower-cost regions – the WAN endpoint fetching from the server is located in the low-cost region, so most content is fetched at lower cost. Only the compressed data between the WAN accelerators enters the developing region.

We intend to use our background in developing content distribution networks [15, 23] to address the geolocation and peer selection mechanisms needed to determine which low-cost region should be used when fetching content. We have some experience understanding the interaction between CDNs and localized content [14, 18], and intend to use this to minimize the possible disruption in choosing off-continent servers to fetch content. One possible casualty of this approach would be localized advertising – if all content is fetched off-continent, then the ads presented may be in the regions of cheap bandwidth, rather than the local region. While this may seem to be of little consequence, many Web sites are ad-supported, and local advertisers may not be willing to patronize content providers that appear to have a mostly foreign audience. Our current plan is to pass along an X-Forwarded-For header that contains the local IP address, and hope that web sites collect that information for advertising and analytics.

## 2.4 Prefetching

While bandwidth-shifting moves bandwidth consumption in space, prefetching moves bandwidth consumption in time. The combination of a Web cache and a WAN accelerator also means that most Web content has some potential utility, either as a fully cacheable page, or as fragments of a page that can populate the WAN accelerator. Off-peak demand, especially for schools, can be near zero, and presents an opportunity to pre-load content for the next day. Traditional concerns regarding prefetching, such as self-interference on the network, are mitigated during off-peak hours. Even the question of utility of prefetched content becomes less important, since having several months of storage capacity makes it unlikely that a prefetch will evict anything recently-used. Even simple prefetching approaches, such as crawling news sites every morning, are likely to have a benefit in shifting bandwidth demand. More complicated approaches, such as analyzing the previous day's traffic logs, are also possible.

The main challenge we expect is to determine how to prefetch previously-unseen content. News sites and aggregator sites can contain dozens or hundreds of links, and for prefetching to be truly useful, we must expect to prefetch all of the images and associated content on the pages of interest. If downloading several dozen to hundreds of full pages per site is not realistic, some mechanism must be used to determine which of these links are most likely to be used in the future. Most existing prefetching research assumes that object popularity and correlation are inputs to the system, whereas in our problem, even the objects themselves are new. The solution to this problem may not be entirely technical – we may opt to visually distinguish links that are already cached, so that the user can see which news stories are likely to load the fastest.

Note that neither prefetching nor bandwidth-shifting actually reduce total bandwidth consumption – they simply move the bandwidth consumption in time or in place, and are likely to actually increase the total number of bits transferred in the network. The benefit, however, is that the number of bits traversing the bottleneck link at peak hours should decrease substantially, improving end-user response time. In environments where every bit is metered, the degree of congestion reduction must be weighed against the added cost. In such environments, much more conservative prefetching may be appropriate.

## 2.5 Snooping

Snooping is an extension to prefetching, and involves using broadcast channels to populate peer caches. Users within wireless range of each other may opt to disable encryption to make their traffic cacheable to other users. Likewise, if multiple schools share the same satellite infrastructure, they may opt to let others populate their caches using the broadcast traffic already consuming bandwidth. A similar idea was used for static caches by the now-defunct commercial service Edgix many years ago, but was done without WAN acceleration, so it only benefited static content. In that system, the satellite broadcast system was used as a feeder to the caches, rather than satisfying two-way traffic for users.

Given the use of WAN acceleration, even fragments of content can provide benefit. In wireless multi-hop mesh network environments, such as those being deployed by OLPC or Meraki, it may be useful to deploy caches at multiple locations in the network to save wireless bandwidth. Snooping on large transfers has been shown to be a problem in these environments [7], but with WAN accelerators, even if part of the transfer is missed when snooping, the rest can be cached.

## 2.6 Offline Access

Transparent off-line access similarly builds around caching and WAN acceleration – when external connectivity is lost, the local Web cache can satisfy requests, but by itself, cannot provide the illusion of full connectivity. However, when combined with the WAN accelerator, it can be used to store multiple versions of dynamic content, such as keeping track of what page was last served to each user. Since WAN accelerators can identify the same content in multiple responses, it forms the basis of a deduplication system – for a dynamic page, one copy of the common content is stored, as well as one content of each set of per-user differences. For pages that are dynamically-generated but contain no indication of per-user customization, we may opt to provide them when they are not available. This approach has been used for several years by the Coral CDN [9] to offload flash crowds, reducing concerns about private information leakage.

In the US and Europe, offline access is likely to be desirable more for personal usage, to allow users to see previously-visited content while out of network range. In contrast, the feedback we have received from our current HashCache deployments is that any content that can be provided offline would be greatly beneficial, given how often external network access is lost. Similar to how we envision modifying links to indicate prefetched content, we may have to use other interface-focused approaches to indicate what content is available offline and how stale it is.

## 2.7 Local Search

Offline access can also be augmented by adding local search support, so that cached content can be searched when online search is unavailable or slow. Our intended model is a blend of Tek [22] and RuralCafe [6] – existing search results are presented when available, but if not, a local search is performed using an embedded search engine. This search engine does not have to perform as well as commercial search services to be useful, since the goal is to still have some content availability during disconnection, even if the specific ranking and presentation is not as polished.

To that end, we may opt to provide less accurate search if it can bound the complexity of indexing. For example, one of the mechanisms that HashCache uses to reduce disk access is to limit the size of hash bins – with a fixed allocation per bin, the disk layout becomes more regular. Search engines have to build inverted indexes, showing which Web pages contain particular words [4]. If offline search can ignore some pages from the indexes for frequently-occurring words, it may be possible to build more disk-friendly indexing data structures.

# 3. RELATED WORK

The spectrum of connectivity options for developing regions ranges from physical distribution of data with intermittent access, such as delay-tolerant networking, to wireless connectivity for rural and remote regions [16, 24]. We view our approach as a later-stage development after the introduction of connectivity, and we focus on those with network access as our initial target.

Much research in WAN optimization has focused on protocol-independent schemes, beginning with Spring and Wetherall [19] finding anchors within packets to recent work extending packet caching to routers [2]. The most closely related work in this area is RTS-id [1], which eliminates redundancy in the wireless environment, by caching recently transferred packets through eavesdropping. However, these approaches work on a per packet basis on link layer, which is less desirable when using low-end disks as the storage facility. In this scenario, our approach of operating on byte streams allows for larger contiguous regions, which reduces the number of disk seeks needed to serve content.

In the area of prefetching, the early work by Pirolli and Pitkow [17] examine log files to understand the behavior of users traversing a Web site. This kind of analysis dominates the prefetching world, with both theoretical approaches [5] and implementations [12] focusing on how to improve prefetching given a known set of links and the past behavior of other users. For our environment, the goal is to prefetch previously-unseen content and move it to our virtually-unlimited disk storage. As a result, the prefetching techniques we develop will have to predict the utility of new content, rather than the usage behavior of existing content. Given that the wide-area link is expected to be much slower or more contended than disk, we do not expect to perform much link prefetching from disk. HashCache already performs some grouping and prefetching of related objects, which reduces the number of seeks for fetching Web pages from disk.

# 4. CONCLUSION

We believe providing first-class Internet access for developing regions is not only crucial for true information exchange and participation, but that it is also technically viable with recent hardware innovations. We argue that the key to first-class access lies in an intelligent networking software stack, which scalably and transparently amplifies the limited physical network capacity beginning to become available in these regions. Scalable caching storage allows practically unlimited storage of past network traffic and WAN acceleration transforms redundant remote delivery to efficient local fetch by eliminating duplicate network transfers. Network traffic cost optimization is possible with careful bandwidth shifting and usage-aware prefetching will enable time shift of bandwidth consumption and maximizig the physical network capacity even when it is idle. Snooping of broadcast wireless traffic can further reduce network traffic without special coordination.

In order to realize our vision, we have developed HashCache, a scalable caching storage system, and Wanax, a high-performance WAN acceleration engine which can comfortably run on a low-powered commodity laptop at the same time. Our early evaluation with realistic workloads shows a promising result in this direction. We are currently developing other techniques on top of these systems and are actively pursuing deployment in developing regions in collaboration with organizations participating in Intel's World-ahead program and the One-Laptop-Per-Child project.

## 5. ACKNOWLEDGMENT

We thank anonymous CFI reviewers for their helpful comments. This research was partially supported by NSF awards CNS-0615237 and CNS-0916204, and KAIST award G04100004.

## 6. REFERENCES

[1] Mikhail Afanasyev, David G. Andersen, and Alex C. Snoeren. Efficiency through eavesdropping: Link-layer packet caching. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[2] Ashok Anand, Archit Gupta, Aditya Akella, Srinivasan Seshan, and Scott Shenker. Packet caches on routers: The implications of universal redundant traffic elimination. In *SIGCOMM*, 2008.

[3] Anirudh Badam, KyoungSoo Park, Vivek Pai, and Larry Peterson. Hashcache: Cache storage for the next billion. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, volume 30, pages 107–117, 1998.

[5] Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. A study of integrated prefetching and caching strategies. In *Proceedings of the ACM SIGMETRICS'95*, 1995.

[6] Jay Chen, Lakshminarayanan Subramanian, and Jinyang Li. RuralCafe: Enhancing web search in intermittent networks. In *Proceedings of WWW'09*, 2009.

[7] Fahad Dogar, Amar Phanishayee, Himabindu Pucha, Olatunji Ruwase, and David Andersen. Ditto - A System for Opportunistic Caching in Multi-hop Wireless Mesh Networks. In *Proceedings of ACM MobiCom*, 2008.

[8] The Measurement Factory. `http://www.web-polygraph.org/docs/workloads/polymix-4/`.

[9] Michael J. Freedman, Eric Freudenthal, and David Mazieres. Democratizing content publication with coral. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[10] Sunghwan Ihm, KyoungSoo Park, and Vivek S. Pai. Wide-area network acceleration for the developing world. In *Proceedings of the USENIX Annual Technical Conference (USENIX'10)*, June 2010.

[11] Intel. Classmate PC, `http://www.classmatepc.com/`.

[12] Ravi Kokku, Praveen Yalagandula, Arun Venkataramani, and Mike Dahlin. NPS: A Non-interfering Deployable Web Prefetching System. In *Proceedings of USITS'03*, 2003.

[13] One Laptop Per Child. `http://www.laptop.org/`.

[14] KyoungSoo Park, Vivek Pai, Larry Peterson, and Zhe Wang. CoDNS: Improving dns performance and reliability via cooperative lookups. In *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[15] KyoungSoo Park and Vivek S. Pai. Scale and performance in the CoBlitz large-file distribution service. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2006.

[16] Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[17] Peter L. T. Pirolli and James E. Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterizations. *World Wide Web*, 2(1-2):29–45, 1999.

[18] Lindsey Poole and Vivek Pai. ConfiDNS: leveraging scale and history to detect compromise. In *Proceedings of the USENIX Annual Technical Conference (USENIX'08)*, 2008.

[19] Neil T. Spring and David Wetherall. A protocol-independent technique for eliminating redundant network traffic. In *ACM SIGCOMM*, 2000.

[20] Squid. `http://www.squid-cache.org/`.

[21] S. Revi Sterling, John W. O'Brien, and John K. Bennett. Advancement through interactive radio. In *Proceedings of ICTD 2007*, 2007.

[22] William Thies, Janelle Prevost, Tazeen Mahtab, Genevieve T. Cuevas, Saad Shakhshir, Alexandro Artola, Binh D. Vo, Yuliya Litvak, Sheldon Chan, Sid Henderson, Mark Halsey, Libby Levison, and Saman Amarasinghe. Searching the world wide web in low-connectivity communities. In *Proceedings of WWW'02*, 2002.

[23] Limin Wang, KyoungSoo Park, Ruoming Pang, Vivek Pai, and Larry Peterson. Reliability and security in the CoDeeN content distribution network. In *Proceedings of the USENIX Annual Technical Conference (USENIX'04)*, 2004.

[24] WiMAX. `http://www.wimaxforum.org/home/`.