First-Class Access For Developing-World Environments

Vivek S. Pai
Department of Computer Science
Princeton University
vivek@cs.princeton.edu

Sunghwan Ihm
Department of Computer Science
Princeton University
sihm@cs.princeton.edu

Anirudh Badam
Department of Computer Science
Princeton University
abadam@cs.princeton.edu

KyoungSoo Park
Department of Computer Science
University of Pittsburgh
kyoungsoo@cs.pitt.edu

1. OVERVIEW

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 [10]. 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 [4, 5]. Long-range wireless can bring connectivity where no connectivity existed [8]. 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 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, and off-line access.

At the heart of our networking stack is a caching storage system called HashCache [1], 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.

On top of HashCache, we layer a WAN accelerator (network packet cache) designed for disks with large capacities but low seek rates. 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. Our WAN accelerator also provides a peering protocol, allowing content to be fetched from closer peers when possible.

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.

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 [7, 12] 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 [6, 9], and intend to use this to minimize the possible disruption in choosing off-continent servers to fetch content.

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.

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. This 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.

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 [3] to offload flash crowds, reducing concerns about private information leakage.

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 [11] and RuralCafe [2] – 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.

3. REFERENCES

[1] A. Badam, K. Park, V. Pai, and L. Peterson. Hashcache: Cache storage for the next billion. In *Proceedings of NSDI'09*, 2009.

[2] J. Chen, L. Subramanian, and J. Li. RuralCafe: Enhancing web search in intermittent networks. In *Proceedings of WWW'09*, 2009.

[3] M. J. Freedman, E. Freudenthal, and D. Mazieres. Democratizing content publication with coral. In *Proceedings of NSDI'04*, 2004.

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

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

[6] K. Park, V. Pai, L. Peterson, and Z. Wang. CoDNS: Improving dns performance and reliability via cooperative lookups. In *Proceedings of OSDI'04*, 2004.

[7] K. Park and V. S. Pai. Scale and performance in the CoBlitz large-file distribution service. In *Proceedings of NSDI'06*, 2006.

[8] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. In *Proceedings of NSDI'07*, 2007.

[9] L. Poole and V. Pai. ConfiDNS: leveraging scale and history to detect compromise. In *Proceedings of the USENIX ATC'08*, 2008.

[10] S. R. Sterling, J. W. O'Brien, and J. K. Bennett. Advancement through interactive radio. In *Proceedings of ICTD 2007*, 2007.

[11] W. Thies, J. Prevost, T. Mahtab, G. T. Cuevas, S. Shakhshir, A. Artola, B. D. Vo, Y. Litvak, S. Chan, S. Henderson, M. Halsey, L. Levison, and S. Amarasinghe. Searching the world wide web in low-connectivity communities. In *Proceedings of WWW'02*, 2002.

[12] L. Wang, K. Park, R. Pang, V. Pai, and L. Peterson. Reliability and security in the CoDeeN content distribution network. In *Proceedings of the USENIX ATC'04*, 2004.