Editors: Munindar P. Singh • singh@ncsu.edu
Krithi Ramamritham • krithi@cse.iitb.ac.in

The aAQUA Approach

Innovative Web 2.0 Tools for Developing Countries

Krithi Ramamritham, Saurabh Sahni, Malathy Baru, Chaitra Bahuman, Arun Chandran, and Manjiri Joshi • *IIT-Bombay*

Anil Bahuman • *Agrocom, India*

As in many regions of the world, people in rural India often lack access to knowledge that's more readily available to people in urban areas. Although rural telecenters are becoming more common, developing content that's presented in local languages, relevant to users, and delivered in an immediately usable form is a challenge here and in rural areas across the globe. To address this, an agricultural portal for rural farmers in India uses innovative database systems and information retrieval techniques. In so doing, it both improves service and addresses connection costs and constraints.

The Internet is making a huge impact by bridging traditional geographic barriers and enabling new businesses across towns, regions, and countries. If such benefits are to reach people throughout the world, from small town Arkansas to rural India, Internet information must be accessible, affordable, relevant, usable, searchable, and up-to-date. In India, rural users currently depend on Web content developed mostly for the Western world and written in unfamiliar languages. Even assuming it's relevant and usable, delivering the typically rich content demands high bandwidth and persistent connectivity. Consequently, many people in rural areas lack access to the huge knowledge base acquired by scientific development through the centuries. Although telecenters are beginning to dot India's rural landscape, a major barrier remains in terms of developing content that is translated into local languages, relevant to users' needs, and delivered in a form they can immediately use.

With these factors — and the specific needs of Indian farmers — in mind, we developed a database-backended question-and-answer system and agri-portal as part of the research activities at IIT-Bombay's Developmental Informatics Lab. Our system, Almost All Questions Answered (aAQUA; www.aaqua.org), is an online, locally archived repository. Users can access aAQUA using a Web browser or Java-compatible mobile phone to create, view, and manage content in three languages (Hindi, Marathi, and English). Our research goal is to incorporate innovations in

- database query optimization and caching,
- cross-lingual multimedia information storage and retrieval, and
- human-computer interaction.

We also seek novel ways of providing expert assistance to identify and address farmers' problems and facilitate farmer-to-farmer interactions. Although we developed it for agriculture, researchers can configure and customize aAQUA to provide expert advice in education, healthcare, and other key domains for developing populations.

The aAQUA System

aAQUA is a discussion forum that provides answers to questions related to agriculture and animal husbandry. Launched as a collaborative effort by IIT-Bombay, Krishi Vigyan Kendra (Agricultural Sciences Outreach Centre) in Baramati, and Vigyan Ashram in Pabal, the effort attempts to build a bridge between knowledgeable agriculturists and knowledge-seeking

Domain Requirements

Information provided to rural users must be topical, relevant, timely, and immediately usable. This demands that a system understand the questions in context — both in time and space — and quickly provide useful, practical solutions from an archive or from online experts.

Our first step is to capture questions in a natural way. A review of our question-answer database shows that farmers' questions are typically brief and sometimes contain spelling or grammar errors. One reason for this is that farmers often have to use the English alphabet to type words from their local languages. In addition, our users sometimes post questions as scanned text documents or upload questions as pictures or in audio or video format.

Second, aAQUA must retrieve content in our three target languages — Marathi, Hindi, and English — and let users search and select database keywords in all three languages as well. Multilingual categorization is useful for users fluent in two or more languages, which is quite common in India. Because users can expand a single search query into its counterparts in all three languages, they can search in their own language and also retrieve content in the other two.

Third, unlike most online discussion forums and Web portals, a review of our search query logs indicates that our users infrequently employ the search feature. Upon noting this, we added the "Browse by agricultural keywords" feature. When users ask a question, the experts use this tool to find previous answers they can reuse. An expert searching for a powdery mildew disease on tomatoes might first look under "Crops" and then refine the search by "disease symptoms" (see Figure A).

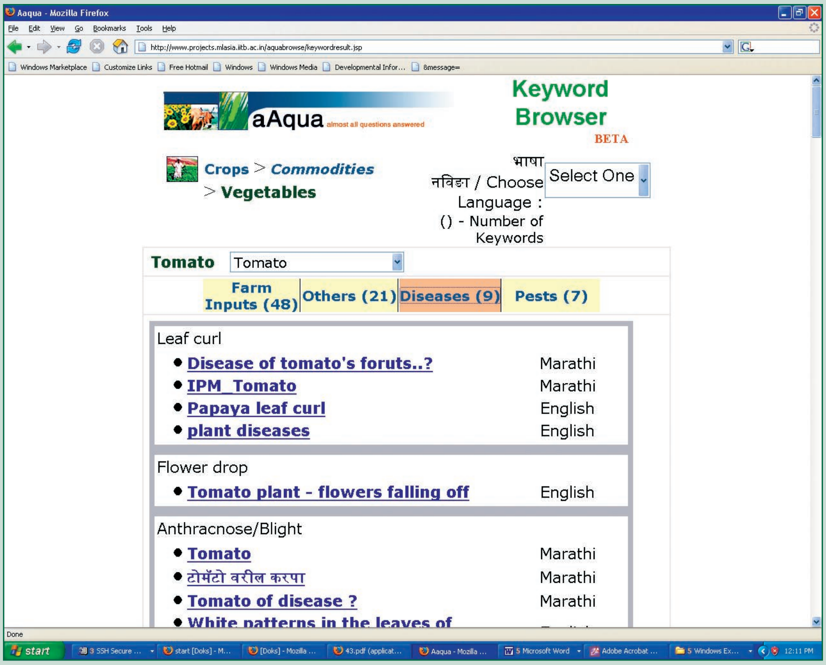

Figure A. Refining a search. The expert searches by crop ("tomato"), then refines the search further by diseases in response to a user's question.

continued on p. 65

users. To accomplish this, we had to address several challenges specific to the rural context (see the "Domain Requirements" sidebar).

In a typical aAQUA thread, a farmer submits a problem, and agriculture experts or other farmers provide solutions (see Figure 1). Currently, users can post a question on the aAQUA site through the Web site, via email, or via mobile texting. As of early February, the aAQUA portal had received 12,052 posts and 626,015 views. To date, 3,925 people have sent new questions; they include a mix of individual farmers, as well as users from farmer organizations, small and medium-sized agribusinesses, and larger companies. As Figure 2 (p. 65) shows, questions have come in from 290 of India's more than 600 districts. Any noncommercial user can browse the forums for free, although users must register on the site before posting a question. Typically, questions come from either farmers or from agri-professionals seeking industrial, financial, or legal advice. The user's profile includes details such as location and weather forecast to offer an appropriate context for the question. Assuming the question is clear and complete, agri-experts provide a detailed answer and attach images or documents if necessary. If the question is incomplete, the agri-expert asks the user to clarify the problem.

Our service is similar to Google Answers and its variants (GA was discontinued in November 2006; for an overview, see http://en.wikipedia.org/wiki/Google_Answers). aAQUA differs from GA in several key respects. First, GA was a paid service, and experts answered only if they accepted the offered bid. Also, there was no guarantee on how long the questioner would have to wait for an answer, although most questions were answered quickly. In contrast, aAQUA experts must reply within 48 hours and compose their answers on the basis of their own acquired knowledge or reference to standard crop- and animal-management practices.

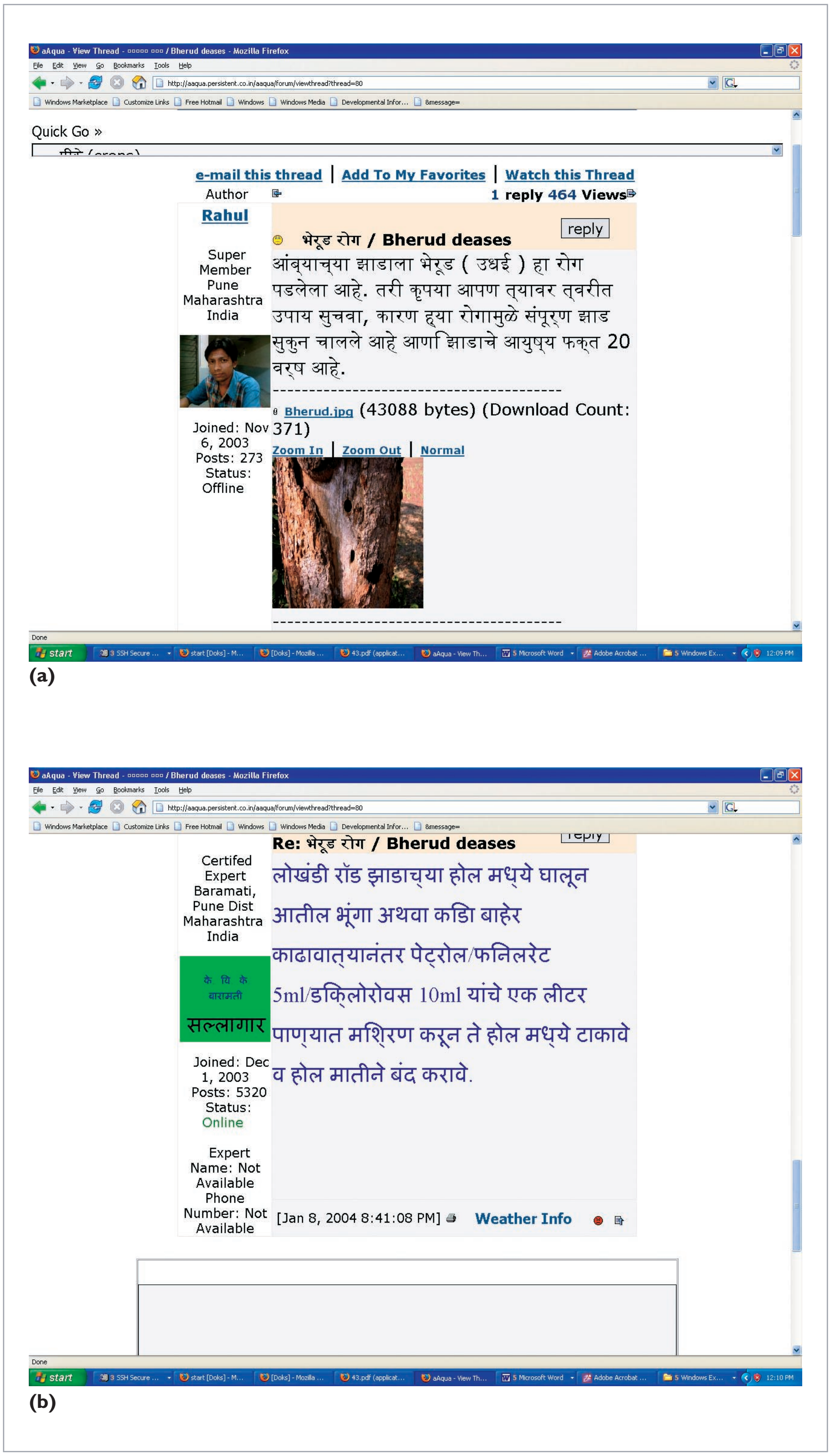

Figure 1. An example aAQUA portal interaction. (a) The question, in Marathi, asks how to deal with Bherud disease, which is drying up a mango tree that's only 20 years old. (b) The expert's response. The green box shows the expert's affiliation (KVK Baramati). The expert advises the farmer to make a hole in the tree and insert an iron rod into it. The insects within the tree will crawl into the rod; the farmer can then remove it and pour a petrol and dichloovus solution into the hole, then cover it with mud.

Basic Technology and Interface

aAQUA uses a three-tier Web architecture with Java Server Pages/Servlets, Oracle, and MySQL databases. We based this on the standard model-view-controller architecture, and run it on a Tomcat Web server. The system uses Unicode UTF-8 compliant databases and Lucene, a Unicode-compliant search and indexing tool.

The aAQUA database comprises mainly tables with attributes for `Member`, `Farmer`, `Expert`, `Moderator`, `Category`, `Forum`, `Posts`, `Thread`, `Permissions`, and `Attachments`. Our target users are predominantly semiliterate non-English speakers who are unfamiliar with the Internet. Our tools provide a simple yet rich interface suitable for new Internet users. We also offer users a Web-based soft keyboard. Experts provide answers to users' questions in the local language (or a combination of languages), paying special attention to the terminology they use and avoiding technical jargon. For example, rather than prescribe quantities measured in parts per million or grams, experts use common measures, such as the teaspoon.

Users typically access our portal through Internet kiosks and cybercafes that commercial enterprises and the Indian government are funding throughout the country. Kiosk operators help semiliterate and illiterate users as well as those unfamiliar with computers. Users at these rural Internet kiosks connect to the Internet on unreliable dial-up connections with low or intermittent bandwidth. Further, they're typically charged by the number of minutes online. To reduce these access costs, we developed a delay-tolerant application, offline aAQUA.

Offline aAQUA Architecture

Offline aAQUA lets users browse and search all aAQUA threads, forums, and other pages without being con-

Domain Requirements (cont. from p. 63)

Fourth, we put all agricultural keywords in the noun form and tag them so that our agri-experts can classify them. Our tagging method is similar to Gmail's labels. To improve document retrieval, we're also incorporating stemmers and spell-checkers (measured by precision and recall). However, precision can be compromised — when, for example, "powdery" is stemmed to "powder," we see a reduction in precision and a considerable increase in recall.

Fifth, we must tailor aAQUA in specific ways to cater to users with limited Internet access. Currently, users can

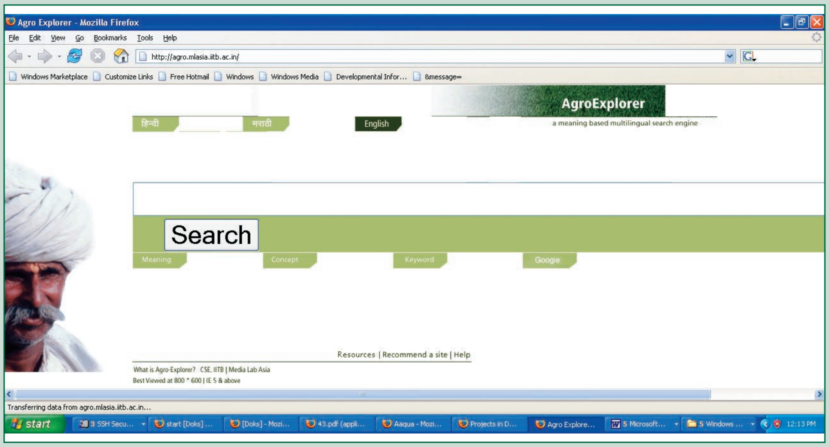

Figure B. Keyword search on aAQUA. This standard search interface is available to all users, whether they access the Web site through a PC or a mobile phone.

- post and receive answers to questions via email and mobile phone text; and
- search aAQUA's documents via mobile phone, either by providing keywords in a standard search interface (see Figure B) or using an organized list of English and Marathi tags in the keyword browser (as in Figure A).

We're also looking at the possibilities of providing rich multimedia applications over mobile phones. We might, for example, use a combination of photos taken with a mobile phone and GPS information to spatially record images of crop infestation. Another example might be to send real-time alerts — such as weather forecasts — to a large number of farmers.

Finally, as the "Offline aAQUA Architecture" section in the main text describes, we addressed users' often slow and discontinuous Internet access by incorporating an offline client that mimics the online user experience.

nected to the site. Because the content is stored on users' computers — and thereby avoids network delays — users can search and browse quickly. Offline aAQUA's local cache or repository also updates whenever users connect to the Internet. The site sends users incremental updates; only the delta is transferred between the clients and the server. If connectivity breaks between thread downloads, users can resume the download later. Users can also post updates to an aAQUA forum in offline mode. The system saves the update on the user's machine and sends it to aAQUA server whenever the client node connects.

Figure 3 shows the offline aAQUA architecture, which is based on heterogeneous database synchronization. The system stores a subset of the server's database on the offline client. The aAQUA portal mainly consists of a threads collection. On the server, our database stores the complete post, metadata, and all other required information for each thread. On the clients, we store only thread metadata — such as thread ID, thread subject, author name, last author, last modified date, and number of replies — in a lightweight database. This lets the system operate efficiently on low-end machines.

We store complete posts separately in a repository (local cache) on the client system, thus reducing the database load. We also index metadata and posts separately to achieve advanced content search capabilities. Users can perform a keyword-based search in all aAQUA threads, as well as do an advanced search on specific attributes such as

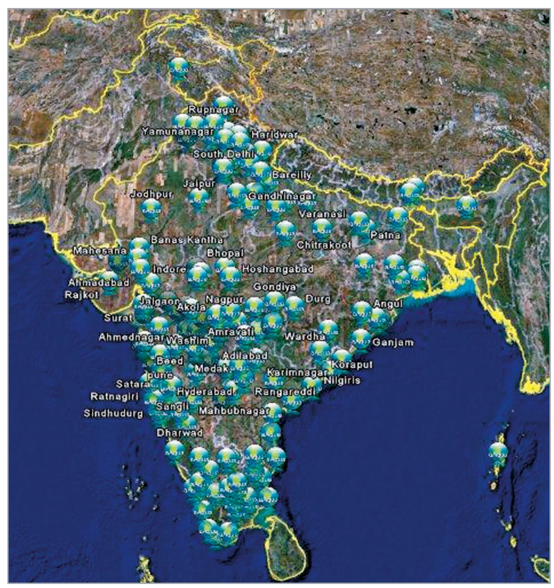

Figure 2. aAQUA system users. Questions for aAQUA come from farmers and agribusiness workers across India.

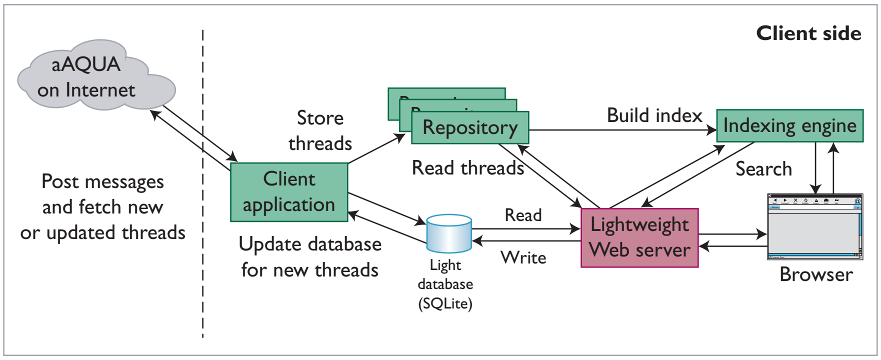

Figure 3. Offline aAQUA architecture. The server database has the complete post, metadata, and all other required information, whereas the light databases on the client systems store only thread metadata. Posts are stored in the repository.

author or last post date. During synchronization, the client first fetches all the server threads that were updated after the client's last update timestamp and then transfers any pending questions, posts, or form submissions to the server.

aAQUA Performance

To measure offline aAQUA's efficacy, we conducted a study to determine the amount of network bandwidth reduction users achieved. We also compared the connectivity time required for online and offline aAQUA. We collected data from the online aAQUA Web site, analyzing 594 visits over a five-day period using both server and client logs. When we saw several hits from a single IP address in one session, we considered that part of one visit. Our estimated data transfer duration is for dial-up connections, with an average download or upload speed of 2.89 Kbytes per second (23.12 kbps) and bandwidth of 56 kbps. Figure 4 offers an overview of how our data transfer rates compare to other systems that offer offline access.

Online Data Transfer

Most of the aAQUA Web site's users view threads, or search or browse forums and other Web site pages. Some users also create posts to ask the experts questions. The average aAQUA visitor session's data transfer is about 351 Kbytes. To create a new post, users typically open the following pages — the homepage, login page, create a new post page, and the post submission page — and are then redirected to the created thread page. On average, these five pages incur a data transfer of 500.25 Kbytes.

Apart from time spent in data transfer, users spend a significant amount of time viewing pages or typing posts. This *elapsed time* can be costly — particularly with dial-up users who are charged by the hour. Because our users can do most of their work offline, we've reduced this elapsed time to zero and thus helped them significantly reduce Internet access costs.

Offline Data Transfer

Offline aAQUA stores all aAQUA pages and the search index locally, so the data transfer involved in browsing is also zero. As Figure 5 shows, transferring a post submitted in disconnected mode to the server incurs an average data transfer of 12.63 Kbytes. The network connectivity duration required for sending a post is 4.37 seconds. We also have to synchronize all online aAQUA updates with offline aAQUA, so it downloads the new and updated threads and stores them locally. Downloading one thread requires an average of 14.3 Kbytes of data transfer, which requires 4.94 seconds of network connectivity.

Offline Browsing

We've explored various possible strategies for offline browsing. A naive method might be to cache a Web site using wget (www.gnu.org/software/wget). However, this requires that users repeatedly download the whole Web site, which is very inefficient. In our current connectivity scenario, even a single such download might not complete at all. Another simple way to realize offline browsing is to use Internet Explorer's synchronization option to cache a Web site locally, and then use Google Desktop Search to search the content offline.

However, interactive Web sites such as aAQUA require that end users post forms, send messages, and so on. Both of these simpler approaches let end users statically browse the data, but not send completed forms to the servers. Also, neither approach lets users fetch only new updates; rather, they must download the entire content each time. Fetching pages from a frequently updated Web site and keeping those pages current would be quite costly using either of these approaches. As we now describe, researchers have proposed models that are more sophisticated. Although they're better and more efficient than the simple models, these approaches nonetheless fail to meet our requirements.

Fish-search-based caching. Researchers have based most of the existing offline browsing techniques on caching pages by *fish search*.[1] In this approach, when users request a Web site download, the system first downloads the starting page and then recursively retrieves links from the downloaded page. The complete Web site downloads only once; a crawler later identifies and fetches new or updated pages. The method specifies the downloaded page set using well-defined boundary conditions and rules, which might be based on parameters such as crawling depth, repository size, and number of pages.

This approach uses a breadth-first search crawl. While pages download, the method incrementally builds a search index to allow offline search. Users can also submit forms and post messages in offline mode by storing posts in XML files, which the system sends to the server when connectivity is established. Fish-search-based caching's advantage over wget and Internet Explorer's synchronization is that it can download only updated pages at a cost of a slightly more complex system. It's a good approach for implementing a generic offline browsing solution for Web sites with frequently updated pages. However, the approach does limit the depth to which you can download a Web site's embedded pages. The method also fails to identify fragments common to different pages and thus redownloads them with every page.

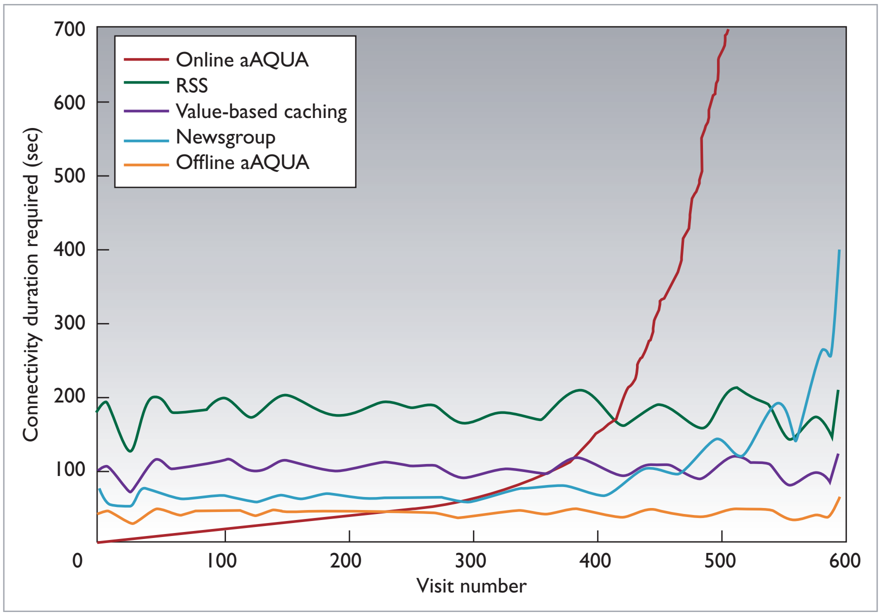

Figure 4. Data transfer rates. How on- and offline aAQUA connectivity times compare to other systems. Offline aAQUA clearly outperforms all other approaches, including value-based caching, RSS, and newsgroup.

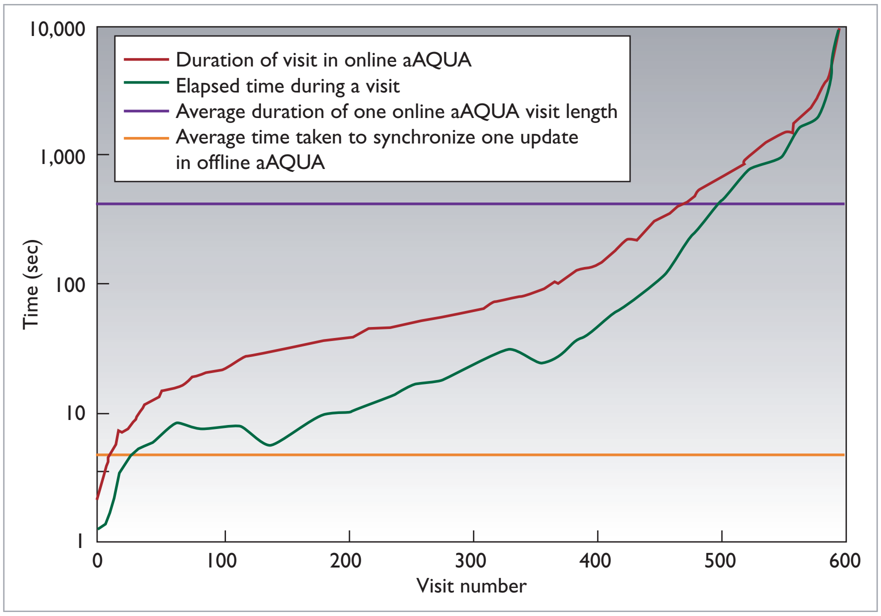

Figure 5. Comparing online and offline data transfer and connectivity times. The results assume a data transfer speed of 2.89 Kbytes per second for different tasks. In addition to data transfer time, users spend a significant amount of time viewing pages or typing posts. Such activities are also included in the required network connectivity time.

Digital-library software. Greenstone (www.greenstone.org) is an open source digital-library software that organizes Web pages and documents in various ways. Once they've added files to Greenstone, users can browse them online as well as offline. When files are added or updated at the server, Greenstone rebuilds its index and transfers it — and the related files — to the local machine. The index transfer's high payload means that sending small updates is quite expensive. It's thus preferable to use Greenstone when you can represent a Web site's content in digital-library form and create large batch updates. Because aAQUA updates are small and quite frequent, this approach is unsuitable for our system.

Homogeneous database replication. In this approach, developers port Web site and database mirrors on a machine closer to end users. The system synchronizes the replicated databases and other files at regular intervals, or whenever Internet connectivity is available. During synchronization, it resolves conflicts and query logs (stored at either end); after synchronization, both databases contain same set of information.

Related Work in Offline Browsing

Many researchers have used various techniques for browsing in disconnected mode. Sean Rhea and colleagues proposed value-based caching,[1] which eliminates redundant data transfer by breaking a page's contents into blocks. The system serves repetitive requests for the same blocks by sending a hash of those blocks. This technique uses low bandwidth efficiently, but requires a proxy to store soft state for each client. All block hashes with each client must be stored at the proxy.

Henry Chang and colleagues proposed and implemented a request-queuing system for disconnected and asynchronous browsing.[2] Whenever a browser sends a request and receives a response, the system caches static pages. Users can then browse and send requests to the cache in disconnected mode. Then, when users establish connectivity, the system serves the queued requests. This approach doesn't let users prefetch a complete Web site for future offline usage; after cache expiry, users must redownload the complete page from the Internet. Also, this approach doesn't work for caching dynamic pages.

Webaroo (www.webaroo.com), a free software service, lets users download any Web site pages that they can access or search without a connection. It also regularly fetches updates for the pages. This approach faces problems similar to content-based caching.

Another model, Web Intelligent Query, implements a proxy server system to minimize bandwidth usage.[3] With WebIQ, users register queries for online search. The system disconnects itself and later reconnects and fetches search results for users to browse offline. However, users can't download complete Web sites for offline access. Users can create an offline forum using an email or RSS or news reader. All new forum posts arrive as emails; by replying to those emails, users post messages to the forum.

With WebIQ, both email and RSS require users to include all page requisites — such as header, footer, and common images — with every email, thereby increasing the payload. Also, although a news reader can receive new forum posts, it can't facilitate offline browsing for other aAQUA portal pages, such as polls, aAQUA keyword browser, and the crop doctor. Moreover, we designed aAQUA's Web-based interface with simple, yet rich textual, graphic, and audiovisual elements for non-English speakers, semi-literate users, and those new to the Internet. Such users would find a news reader's interface quite unfriendly and difficult to use.

Apart from these systems, there are several commercially available offline browsers, including offline explorer (www.metaproducts.com/OE.html), WebCopier (www.maximumsoft.com), WebZip (www.spidersoft.com), BackStreet Browser (www.spadixbd.com/backstreet), WebStripper (www.webstripper.net), and Grab-a-Site (www.bluesquirrel.com/products/grabasite). To keep the local cache and server content synchronized, all of these products must repeatedly download the entire Web site, which is highly inefficient. Bandwidth limits the depth to which users can download a Web site's pages. Because the system will repeatedly download and store common fragments of different pages, it wastes both network resources and disk space. With

continued on p. 69

Table 1. aAQUA goals and outcomes.

aAQUA goals	Current status
Reachable	Deployed across 290 out of approximately 600 districts in India
Accessible	Users can access it via PCs and mobile devices, including those with limited or intermittent Internet access
Affordable	Kiosk operators charge less than 10 rupees per question (approximately 25 cents)
Relevant	Content is customized to users' specific questions
Usable	Most users and extension staff find it easy to use
Searchable	Users can search by keyword selection or keyword search
Up-to-date	Experts provide answers within 36 hours; archives are updated

Homogeneous database replication can provide an excellent offline experience with local-server connectivity. It also has a small synchronization payload and improves response time and robustness in content delivery. The method's disadvantage is that deploying it requires the client to have dedicated, high-performance databases, which in turn requires administrators to manage them. This approach is useful when a Web site has many users in a local region, such as within a LAN.

Table 1 shows our outcomes in relation to our initial goals. The many-fold reduction in bandwidth that we achieve using offline aAQUA makes it well suited to the requirements of developing regions. Our approach is also useful for caching other delay-tolerant dynamic Web sites for browsing and searching in disconnected mode. To achieve this, Web site administrators must iden-

Related Work in Offline Browsing (cont.)

this approach, only small, rarely updated Web sites can be cached efficiently.

No existing solution effectively addresses the problem of bandwidth waste due to a system repeatedly downloading and storing common fragments. In a forum like aAQUA, an addition or update to a single thread changes several pages — when a new thread is added, for example, it changes all pages that list recent threads in reverse chronological order. Figure C shows an added thread (circled on right-hand page) that bumps what was the page's final thread (circled at bottom left) to the second thread-listing page. This movement repeats across all thread-listing pages, updating them all. Here, even techniques that permit downloading only updated pages will prove a failure. We therefore need a solution that can optimize network and disk storage usage, transfer only the "delta" of updates, and yet also present consistent results to the user. To achieve this, the offline system must take on a certain amount of page construction.

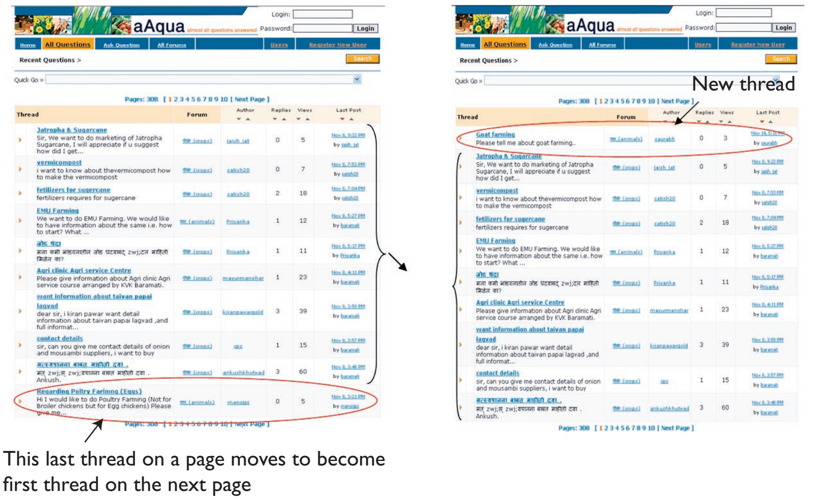

Figure C. Adding a thread. When a thread is added to a page, the final entry shifts to the next page, and so on across all thread-listing pages.

References

1. S.C. Rhea, K. Liang, and E. Brewer, "Value Based Web Caching," *Proc. 12th Int'l Conf. World Wide Web*, ACM Press, 2003, pp. 619–628.
2. H. Chang et al., "Web Browsing in a Wireless Environment: Disconnected and Asynchronous Operation in Artour Web Express," *Proc. 3rd Int'l Conf. Mobile Computing and Networking* (MobiCom 97), ACM Press, 1997, pp. 260–269.
3. R. Kavasseri et al., "Web Intelligent Query-Disconnected Web Browsing Using Cooperative Techniques," *Proc. 1st IFCIS Int'l Conf. Cooperative Information Systems* (COOPIS 96), IEEE CS Press, 1996, pp. 167–174.

tify the dynamic and static content to make it available offline. The site's offline installer can ship static pages and fragments, whereas dynamic content can be updated during synchronization. Developers can design schema for the offline database by analyzing the dynamic fragments and pages. The synchronization process and changes required at the Web site server are similar to those of offline aAQUA.

As for future work, there are many avenues ahead. First, we'd like to experiment with using the Vector Space Model[2] to generate profile and context information in aAQUA with minimum user intervention. By understanding usage patterns, the system could prefetch pages that match a profile. For example, the system might prefetch all updated threads related to the onion crop for users interested in onions.

Second, we're planning to develop mechanisms that let users easily collaborate with peers and share information. The clients must be highly interactive, offering data and services to their peers, while at the same time remaining autonomous so the local cache can serve most requests and facilitate successful collaboration.[2,3] In offline aAQUA, one user might cache all threads related to "onion" and "potato," while another peer caches those related to "onion" and "cucumber." Collaboration would let both users browse threads related to all three topics. Also, only one user would need to receive onion-thread updates to make them available for both to download. A server's data coherency is always higher compared to that of a peer. Deciding whether to download from a peer or the Internet would rest on several parameters, such as download time, the target page's lifetime characteristics, how long the peer had the page, and fidelity requirements.

We have several other research possibilities, including

- using the benefits of novelty detection and automatic fragmentation to further reduce the data transfer payload and provide users even better performance;
- exploring how to use offline aAQUA on palmtops and other mobile devices; and
- examining when an offline aAQUA client should synchronize to provide optimal user respon-

siveness using any one of several existing techniques.[4,5]

Currently, Agrocom, a spin-off startup company based at IIT Bombay, is helping us replicate aAQUA within the Indian Agri-University and Agri-Extension System. Specifically, Agrocom is increasing the number of experts on the panel and creating partnerships with organizations that serve numerous farmers. We currently have the ambitious goal of reaching 1 million farmers by 2011.

References

1. P.M.E. De Bra and R.D.J. Post, "Searching for Arbitrary Information in the WWW: The Fish-Search for Mosaic," *Proc. 2nd Int'l Conf. WWW,* ACM Press, 1994.
2. F. Perich et al., "Collaborative Joins in a Pervasive Computing Environment," *Very Large Databases J.*, vol. 14, no. 2, 2005, pp. 182–196.
3. F. Perich et al., "On Data Management in Pervasive Computing Environments," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 5, 2004, pp. 621–634.
4. S. Garg, K. Ramamritham, and S. Chakrabarti, "Web-CAM: Monitoring the Dynamic Web to Respond to Continual Queries," *Proc. Int'l Conf. Management of Data*, ACM Press, 2004, pp. 927–992.
5. S. Pandey, K. Dhamdhere, and C. Olston, "WIC: A General Purpose Algorithm for Monitoring Web Information Sources," *Proc. 30th Int'l Conf. Very Large Databases*, VLDB Endowment, 2004, pp. 360–371.

Krithi Ramamritham is a professor of computer science and engineering and the dean of Research and Development at IIT-Bombay, and principal investigator of the Developmental Informatics Lab. His interests include database systems, real-time systems, and Internet computing. Ramamritham has a PhD in computer science from the University of Utah. He is a fellow of the IEEE, the ACM, and the Indian National Academy of Engineering. Contact him at krithi@iitb.ac.in.

Anil Bahuman is a social entrepreneur who recently cofounded Agrocom along with Krithi Ramamritham and Bishnu Pradhan at the SINE Business Incubator at IIT-Bombay. His research interests include wireless networks and developmental informatics. Bahuman has a bachelor's degree in electrical engineering from NIT-Surathkal and an MS in artificial intelligence from the University of Georgia. He serves as a member of the board at Vigyan Ashram, Nikshepa Technologies and is the chairman of the Weather Insurance Guidelines and Standards Forum (WIGS). Contact him at abahuman@it.iitb.ac.in.

Saurabh Sahni is a senior software engineer with Yahoo! India Research & Development, Bangalore. His research interests include dynamic data dissemination mobile environments, algorithms, Internet technologies, and network security. Sahni has a BS in electronics engineering from IET-DAVV, Indore. Contact him at saurabh.sahni@gmail.com.

Malathy Baru is a project manager at the Developmental Informatics Lab, IIT-Bombay. Her research interests include information and communication technologies and human-computer interface design. Baru has a BS in electronics engineering from Osmania University, Hyderabad. Contact her at malati@it.iitb.ac.in.

Chaitra Bahuman is a senior developer at the Developmental Informatics Lab, IIT-Bombay, and currently heads a team working in the area of content delivery over mobile phones for agriculturists in rural Maharashtra. Her research interests include emerging technologies and mobile computing. Bahuman has an MS in computer applications from Visweswariah Technological University, Karnataka. Contact her at chaitra@it.iitb.ac.in.

Arun Chandran is a senior developer at the Developmental Informatics Lab, IIT-Bombay. His research interests include database systems and Internet technologies. Chandran has an MS in computer applications from Amrita University, Kerala. Contact him at arunc@it.iitb.ac.in.

Manjiri Joshi is a research assistant at the Developmental Informatics Lab, IIT-Bombay. Her research interests include Internet computing and mobile computing. Joshi has a BS in computer science and engineering from Nanded University, Maharashtra. Contact her at manjirij@it.iitb.ac.in.