From the Editor: Web Systems Community—To Those Who Believe We are Done Already

Guillaume Pierre, Vrije Universiteit

Friends can sometimes be the harshest reviewers. The torture usually starts during a dinner party or other pleasant get together. One of them asks, "Could you tell me once more what your job is about?" At that point, it's already too late. Many of my friends aren't computer scientists, so I typically answer something along the lines of, "Well, I am doing research to improve the Web." Inevitably, my friends start telling me how great the Web is and how they couldn't live without it. But then it often strikes them: since the Web is so great, what's left to research?

I don't blame my friends for lacking the very special kind of imagination necessary to identify how the Web isn't so great and what should be done about it. But I do find it surprising when researchers express similar opinions. So, to those who believe we are done already, here are a few of the Web's wrong or missing features.

**Flash crowds**

Take an ordinary Web server—for the sake of the explanation, the Belgian biosafety agency's server. From a technical standpoint, this server is perfectly unremarkable. To be more precise, that was the case until the bird flu crisis hit. On 31 October 2005, the Belgian government decided that all chickens should be kept indoors, not only in industrial settings but also in individual farmyards. This measure applied only to specific risk areas, which you could find listed on the agency's Web site. Hours later, Belgian news headlines featured apologies from agency officials about the server's unavailability and their promises that it would be reinforced as soon as possible.
The sudden and huge increase in request rates the server experienced is known to Web administrators as a *flash crowd*. Within a few minutes or even seconds, request traffic grows so much that it exceeds the server's capacity. In extreme cases, servers can crash or even be physically damaged. Flash crowds happen for a variety of reasons, ranging from people seeking information on a major news event, to a new hypertext link on a popular bulletin board (http://en.wikipedia.org/wiki/Slashdot_effect), to a poorly planned marketing campaign (http://www.cnn.com/TECH/computing/9902/05/viceweb.idg/).

Flash crowds have long been an important research problem, and numerous solutions exist. Organizations can operate Web sites through a cluster of computers instead of a single server, which lets the sites sustain a much larger load. For organizations unwilling to buy and administrate a cluster, commercial content delivery networks will host the content in their own infrastructure. Even better, they'll replicate the site worldwide, so a copy is always nearby any client, greatly improving download performance. Peer-to-peer systems such as BitTorrent provide similar functionality by sharing the burden of serving the content with all users currently downloading the content.

Because many solutions exist, it's tempting to consider the problem solved. Yet, most Web servers don't use any of these techniques and consequently fail during flash crowds, precisely when their content becomes relevant to many clients. Is that a satisfactory situation? Obviously not.

Ideally, every Web server would have a "rescue" function that asks others for help when temporarily overloaded. This is technically feasible, as demonstrated, for example, by the DotSlash (http://www.cs.columbia.edu/~zwb/project/dotslash/) project. However, such systems open a new can of worms: How do you locate servers likely to rescue you? Should servers charge each other for using their resources? If not, how can servers confirm that those calling for help are indeed flash-crowded? How do flash-crowded servers confirm that rescue servers are keeping their promises and, for example, not delivering maliciously modified content to the users? All these questions do not receive the attention they deserve.

**Intranet search**

If you don't operate a Web server, you might feel unconcerned about flash crowds. So, we'll take a Web-related activity that you can't avoid: searching. The Web is so big nowadays that finding relevant information requires a search engine, such as Google or MSN Search. You might think, "These systems work great, so what's to research?"
Suffice it to say that searching is one of the major current Web research topics. At the latest International World Wide Web Conference (http://www2005.org/), I counted no fewer than 16 papers published about search.

I was surprised, however, to see no paper whatsoever on intranet search. Now that most companies store so much of their internal information digitally, locating the right information in an intranet becomes increasingly difficult—hence the need for intranet search. But at first glance, searching for documents in an intranet seems easier than across the whole Internet. After all, the Internet is much bigger, isn't it?

Intranet search was the topic of a very interesting panel at the 2003 IEEE Workshop on Internet Applications (http://www.cs.ucdavis.edu/~aksoy/wiapp03/). During the panel discussion, an intriguing case emerged: intranet search is in fact much more difficult than Internet search.

Let me explain. Accessing information over the Internet is realized through just two protocols: HTTP and its secure variant HTTPS. Internet crawlers therefore must implement only HTTP and HTTPS to retrieve and index documents. Similarly, search engines are typically requested by Web browsers, so returning HTTP addresses in response to a search guarantees that browsers will know how to interpret them to access the content. In intranets, only a fraction of the information is accessible via HTTP. Intranets often use a combination of Web sites, shared file systems, and databases to store information. This of course means that intranet crawlers must support a whole variety of access protocols. And, that's not all: search engines should also be able to return references to information that their clients can exploit. It then becomes necessary to build client software that can uniformly access information using the same large variety of access protocols. Can your Web browser directly query an Oracle or a Lotus Notes database? Mine can't.

Indexing information in an intranet is also harder than in the Internet. Internet information is organized in pages, which crawlers can index independently from each other and search engines can return as responses to queries. If information in an intranet is, for example, stored in a relational database, what should be the indexing granularity? Should search queries return a whole database, a table in a database, a record in a table in a database, or something in between? The answer to this question depends to a large extent on the database structure's semantics, which makes it difficult for automated search engines to deal with.

Finally, in the Internet, crawlers usually index only publicly available documents. They generally ignore all access-controlled information. In his white paper "The Deep Web:
Michael Bergman estimates this unindexed content to be about 500 times larger than the indexed content. This is a search engine limitation, but one that most users can easily understand. In an intranet, all relevant information is by definition subject to access control. Ignoring all restricted information in an intranet search engine would negate its usefulness, so intranet crawlers must run with extended access rights. This creates a new problem, because search queries should return only references to information that the requester is allowed to access. For example, a search for "Layoffs 2006" shouldn't return any response unless the requester is authorized to view that information. The sheer existence of such a document is information in itself, so an unauthorized requester should be prohibited not only from viewing the document's contents but also from knowing the document exists.

Two years after some leading experts identified intranet search as a hot research area, I would expect to see extensive research being conducted on the subject or the problem solved and sophisticated products available. I might be mistaken, but I see none of those.

Apart from these two topics, I haven't even mentioned preventing bad behavior such as adversaries stealing information from your computer while you search the Web and companies gradually getting to know all about your shopping behavior. Users often can protect their kids from seeing certain sites only by forbidding all Internet access. In general, we have virtually no means to protect ourselves against or to track down many producers of unsolicited material. The Web has many unsafe places, and they're often difficult, if not impossible, to avoid.

To those who believe we are done already, let me tell you one thing: Web systems research has clearly made immense progress since the Web was invented in 1994, but we're not there yet.

Guillaume Pierre is an assistant professor at Vrije Universiteit, Amsterdam. Contact him at gpierre@cs.vu.nl, or visit his Web site www.cs.vu.nl/~gpierre.
Related Links

DS Online's Web Systems Community, http://dsonline.computer.org/was


Cite this article: Guillaume Pierre, "To Those Who Believe We are Done Already," IEEE Distributed Systems Online, vol. 7, no. 1, 2006, art. no. 0601-o1002.