Summary

Collaborative Wikipedia Hosting

The development of the Internet has facilitated new types of collaboration among users. One of the most interesting results of this user collaboration is Wikipedia, an online encyclopedia based on wiki technology, which allows any user to create and edit the contents of its articles. This model, combined with the noncommercial nature of the project, has attracted hundreds of thousands of individuals that voluntarily contribute content, resulting in the largest and most popular general reference work on the Internet. According to independent statistics, Wikipedia is the seventh most popular Web site in the world.

The operation of a top Web site is a complex and expensive task. In Wikipedia’s case, it is made more difficult by the fact that the continuing increase in its popularity does not necessarily translate into a proportional increase of its income, which is composed basically of donations. This limitation of resources and the noncritical nature of the application has made Wikipedia trade off reliability in favor of performance. This has manifested itself in occasional continental or global-level outages.

Regarding scalability, it can be said that, in principle, Wikipedia’s infrastructure has no obvious faults in any of the traditional scalability dimensions (size, geographical or administrative). Wikipedia engineers have done remarkable work implementing a custom content delivery network with data centers in two separate continents that can be efficiently managed by a small administrative team. However, there is indeed a scalability issue and it is related to the fact that the infrastructure can only grow significantly by adding large, expensive data centers. This is generally not a problem for other large popular commercial Web sites, but in Wikipedia’s case it can become a serious problem because its limited financial resources make it difficult to introduce new data centers. For the same reason, it can
be said that there may be some administrative scalability concerns, as it may be difficult for Wikipedia to assemble a sufficiently large administrative team to deal with the growing complexity of the Web site.

It is our thesis that Wikipedia could significantly improve its reliability and scalability in an economical way by implementing a collaborative hosting infrastructure in which individuals are allowed to contribute some of their unused hardware resources to help host the Web site, similarly to the way they already contribute content or donate money. We base this idea on the fact that thousands of people dedicate nontrivial amounts of time or money just to help Wikipedia. We believe that this willingness to contribute could be extended to an additional collaboration mechanism that would presumably have little cost for the potential contributors.

A practical collaborative hosting system for Wikipedia must satisfy three important functional requirements. First, the existing Wikipedia functionality must not be affected. Everything that can be done with the current Wikipedia, should continue to be doable with a collaboratively hosted one. Second, end-user operation should not be disturbed. This implies that, after the collaborative hosting system is deployed, current Wikipedia users should continue to use Wikipedia as if nothing had happened, using a regular Web browser without any special configuration change. Third, the Wikipedia operator must be able to continue controlling Wikipedia the way they do now, which includes retaining full control over editorial policies and mechanisms such as authentication. In this dissertation we study the problem of how to design a collaborative Web hosting system that satisfies these requirements.

Such a collaborative hosting system must also satisfy a number of non-functional requirements that are common in many decentralized systems. First, it must scale to large numbers of users and participating nodes. Second, it must cope with participating nodes that join and leave the system at any time. Third, it must tolerate possible attacks perpetrated by malicious participating nodes. Fourth, it must deal with trust issues introduced by the participation of nodes controlled by entities that are unknown and untrusted by end users. In particular, it should be impossible for an untrusted participating node to know the history of read operations for any individual user.

To achieve our objective of designing a collaborative hosting system for Wikipedia, we first studied Wikipedia’s workload and current hosting infrastructure comprehensively. To conduct this study, we used a trace that contained a 10% sample of the Wikipedia traffic during a 107-day period.

A decentralized collaborative hosting system for Wikipedia necessarily
requires the use of basic peer-to-peer (P2P) technologies as infrastructure elements. However, many of these technologies are known to be vulnerable to attacks perpetrated by malicious participants. In order to make sure that our hosting system would not depend on knowingly insecure building blocks, we made a comprehensive study of security in P2P systems, where we analyzed and compared many proposed P2P security techniques. From this study we gained insight and derived guidelines on how to use P2P systems in a secure way to build higher-level applications such as our Wikipedia hosting system.

An important conclusion of our workload and security analyses was that a fully decentralized solution based exclusively on P2P technologies would be neither scalable nor secure. Moreover, we found out that the existing Wikipedia architecture, which is based on distributed caches, a multicast-based cache-invalidation system, and centrally maintained data stores, is superior to any of the proposed P2P alternatives, which usually focus on decentralizing the back-end data store.

Our solution to the problem of collaborative Wikipedia hosting is an architecture that combines centralized and decentralized elements and uses a conceptually similar approach to the one used by the existing Wikipedia infrastructure.

In our architecture there are three groups of participants. One is the Wikipedia operator itself, who administers a trusted infrastructure that includes centrally maintained data stores and handles all the functionality that is different from reading wiki pages. This functionality is, in many cases, difficult to implement in a secure and scalable way using P2P technologies, but it represents only a small fraction of the Wikipedia workload. The second group of participants are organizations such as ISPs, universities, and companies, which provide their end users with front ends that behave similarly to standard transparent Web proxies, but that cache and render Wikipedia content using custom protocols that can produce greater benefits for the organization than using a standard Web proxy. Finally, the third group of participants are individuals who contribute part of their computing, storage and networking resources to help host Wikipedia. The role of these users is to act as a secondary cache layer to protect the trusted infrastructure from the read load generated by cache misses in the front-end layer, and to help maintain the consistency of front-end caches. Note that this architecture is similar to the existing Wikipedia infrastructure, but first-level caching is performed by organizations that are close to and trusted by end users, while cache-consistency overhead and second-level caching are largely the responsibility of collaborative users.
With this architecture, the Wikipedia operator is responsible for non-read requests and for read requests that cannot be processed by the first two caching layers. Since Wikipedia’s workload is read mostly, it is highly cacheable and, therefore, the Wikipedia operator needs to handle only infrequent requests.

We validated our proposed architecture using trace-based simulations. We used the same trace that we used for the workload analysis, and an implementation of the architecture that runs on top of a custom discrete-time event-based network simulator. Our experimental evaluation shows that the proposed architecture is able to handle the Wikipedia workload efficiently. The architecture is able to take advantage of the resources contributed by collaborative users to significantly reduce the load that needs to be handled by the trusted infrastructure. The architecture is also able to provide good user-perceived quality of service while enabling bandwidth savings for organizations providing front ends.