Decentralized Search on Decentralized Web

Eric Lo
Department of Computer Science and Engineering
Chinese University of Hong Kong (CUHK)

Decentralized Web, or DWeb, is envisioned as a promising future of the Web. Being decentralized, there are no dedicated web servers in DWeb; Devices that retrieve web contents also serve their cached data to peer devices with straight privacy-preserving mechanisms. The fact that contents in DWeb are distributed, replicated, and decentralized lead to a number of key advantages over the conventional web. These include better resiliency against network partitioning and distributed-denial-of-service attacks (DDoS), and better browsing experiences in terms of shorter latency and higher throughput. Moreover, DWeb provides tamper-proof contents because each content piece is uniquely identified by a cryptographic hash. A DWeb prototype, which hosts a Wikipedia snapshot, can be found here. DWeb also clicks well with future Internet architectures, such as *Named Data Networking* (NDN).

Search engines have been an inseparable element of the Web. Contemporary ("Web 2.0") search engines, however, provide centralized services. They are thus subject to DDoS attacks, insider threat, and ethical issues like search bias and censorship. As the web moves from being centralized to being decentralized, search engines ought to follow. We propose QueenBee, a decentralized search engine for DWeb. QueenBee is so named because worker bees and honeycomb are a common metaphor for distributed architectures, with the queen being the one that holds the colony together.

QueenBee aims to revolutionize the search engine business model by offering incentives to both content providers and peers that participate in QueenBee's page indexing and ranking operations. Figure 1 shows our vision of QueenBee, whose core business operations are autonomously and securely governed by *smart contracts* deployed on a cryptocurrency blockchain like Ethereum.

QueenBee advocates *no-crawling*, because crawling inevitably reduces the freshness of the search results. Instead, QueenBee incentivizes content creators to *publish* (create or update) their contents via QueenBee's smart contract to gain "honey" in the form of a cryptocurrency. Honey is also rewarded to worker bees – peers that help update the index and

9th Biennial Conference on Innovative Data Systems Research (CIDR '19) January 13-16, 2019 , Asilomar, California, USA.

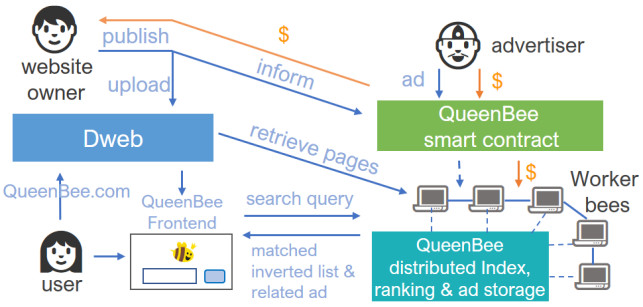

Figure 1: QueenBee and the Dweb

compute the page ranks, which are hosted in a decentralized storage (e.g., IPFS). Users submit their keyword queries via QueenBee's HTML+Javascript frontend on the DWeb. The frontend is also responsible for composing the search results by intersecting the matched inverted lists, ranking the results, and displaying relevant ads.

QueenBee is a *decentralized organization* – advertisers directly make advertisements through our smart contract and *the ad revenue is shared among the content creators and worker bees*. QueenBee's decentralized nature rids itself of problematic issues (e.g., search bias) found in centralized search engines. To our best knowledge, QueenBee is the world first initiative to build a decentralized search engine on the DWeb. Existing P2P search engines (e.g., YaCy) only work on Web 2.0, without an incentive scheme or a security incentive that guard against practical attacks.

Besides performance issues, QueenBee will face many new and interesting research challenges. We briefly discuss two of them. (I) *A fair incentive scheme for all stakeholders*: For example, while allowing any content provider to use our service, we need to reward those whose websites are popular. A simple way is to give the providers for which the page ranks of their websites exceed a certain threshold some QueenBee's honey. For advertisers, we also need a fair scheme to charge them (e.g., they pay by the number of clicks on the ad). In general, a sensible scheme is needed to maintain the ecosystem of QueenBee. (II) *New attacks*: this new model of decentralized search engine may induce new attacks. For examples, an attack from colluded worker bees that aim at manipulating QueenBee's indexes or page ranking data maliciously (*collusion attack*); as popular webpages will gain QueenBee's honey, *scrapper site attack* may exist that tries to mirror popular websites for QueenBee's honey.