**INTRODUCTION**

**Problem Statement**
Most search systems manage Web Crawlers using a centralized client-server model in which the assignment of crawling jobs is managed by a centralized system using centralized repositories. Such systems suffer from a number of problems, including link congestion, low fault tolerance, low scalability, and expensive administration.

**Our Solution**

DSphere (Decentralized Information Sphere) performs crawling, indexing, searching and ranking using a fully decentralized computing architecture.

DSphere has a Peer-to-Peer network layer in which each peer is responsible for crawling specific sets of documents, referred to as the source collection. A source collection may be defined as a set of documents belonging to a particular domain.

Each peer is also responsible for maintaining an index over its crawled collections and ranking its documents using a source-centric view of the web which replaces the page-centric view used by current search engines.

**P2P CRAWLER**

**P2P Web Crawlers**

- **Apollode** - Structured P2P Network
- **PeerCrawl** - Unstructured P2P Network

**Most Important Features**

- **Division of Labor** – Mapping of URLs to peers for crawling. Duplicate mapping has to be avoided as far as possible.
- **Apollode** uses the DHT protocol for distributing the World Wide Web space among all peers in the network.
- **PeerCrawl** performs the division of labor by introducing a hash-based URL Distribution Function that determines the domains to be crawled by a particular peer. The IP address of peers and domains are hashed to the same m-bit space. A URL U is crawled by peer P if its domain lies within the range of peer P. The range of Peer P, denoted by Range(P), is defined by:
  \[ h(P) \times 2^k \text{ to } h(P) + 2^k \]

  where \( h \) is a hash function (like MD5) and \( k \) is a system parameter dependent on the number of peers in the system. In our first prototype of DSphere, we use the number of neighbor peers of \( P \) as the value of \( k \).

**SOURCE RANKING**

DSphere computes two scores: (1) each source is assigned an importance score based on the analysis of the intra-source link structure; and (2) each page within a source is assigned an importance score based on an analysis of intra-source links.

We plan to incorporate a suite of spam-resistant countermeasures into the source-based ranking model to support more robust rankings that are less difficult to manipulate than traditional page-based ranking approaches.