DSphere: A Source-Centric Approach to Crawling, Indexing and Searching the World Wide Web

Distributed Data Intensive Systems Lab | Georgia Institute of Technology

INTRODUCTION

Problem Statement

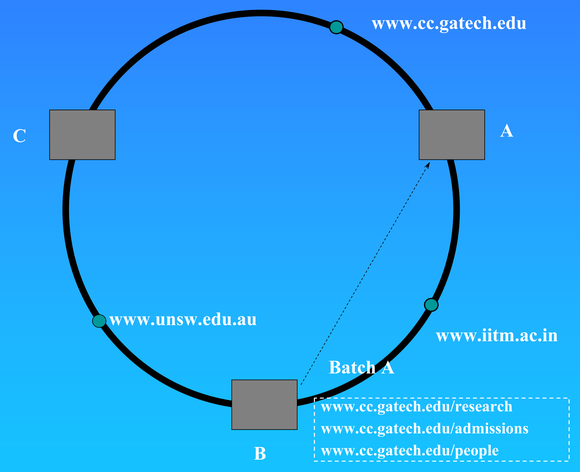
Most search systems manage Web Crawlers using a centralized client-server model in which the assignment of crawling jobs is managed by a centralized system using centralized repositories. Such systems suffer from a number of problems, including link congestion, low fault tolerance, low scalability and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

Our Solution

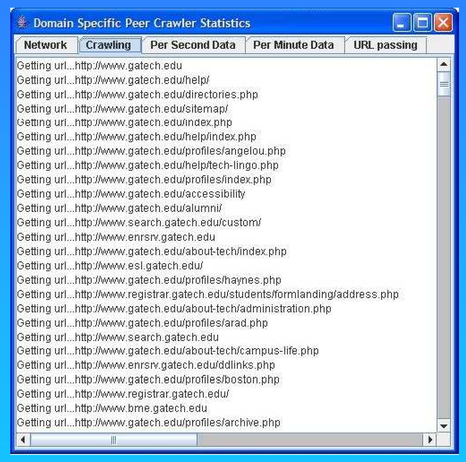
DSphere (Decentralized Information Sphere) performs crawling, indexing, searching and ranking using a fully decentralized computing architecture.

DSphere has a Peer-to-Peer network layer in which each peer is responsible for crawling a specific set 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.



Division of Labor



Screenshot of Crawler

IMPLEMENTATION

- ❑ PeerCrawl uses Gnutella protocol for formation of the network layer
- ❑ *Phex* open source P2P file sharing client
  - ❑ Necessary interfaces for network formation and maintenance.
  - ❑ Current prototype assumes a flat architecture wherein all nodes have equivalent capabilities.
  - ❑ Crawler is built on this layer and makes appropriate calls to Phex routines as needed.
- ❑ Uses local host caching and web caching to connect to P2P network
- ❑ Peer communication
  - ❑ Peers broadcast URLs not in their crawl range
  - ❑ Maintain count of nodes on horizon for dynamic adjustments to the crawl range.
- ❑ Multiple threads performing specific functions
  - ❑ *Fetch_Thread* : Gets document from server. Can use local DNS mappings to expedite process.
  - ❑ *Process_Thread* : Extract URLs from document. Filters URLs based on policies like Robots Exclusion.
  - ❑ *Caching_Thread* : Stores documents to persistent storage. Useful in building a web archive.
  - ❑ *Statistics_Thread* : Maintains book-keeping information.
  - ❑ *Network_Connection_Thread* : Checks with Phex client for peers joining/leaving network.
  - ❑ *Dispatch_Thread* : Broadcasts URLs not in range.
  - ❑ *Backup_Thread* : Periodically backs up data structures.
- ❑ URL duplicate detection
  - ❑ Uses Bloom Filters for detecting duplicate URLs
- ❑ Checkpointing
  - ❑ Allows crawler to restart from last saved state of data structures in case of crashes.

P2P CRAWLER

P2P Web Crawlers

Apoidea – Structured P2P Network
PeerCrawl – Unstructured P2P Network

Most Important Features

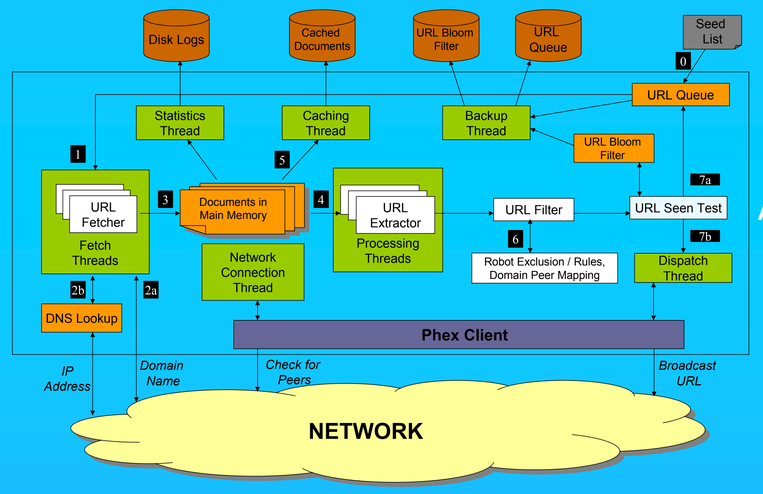
Division of Labor – Mapping of URLs to peers for crawling. Duplicate mapping has to be avoided as far as possible.

Apoidea 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:

$$h(P) - 2^k \text{ to } h(P) + 2^k$$

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*.



Crawler Architecture

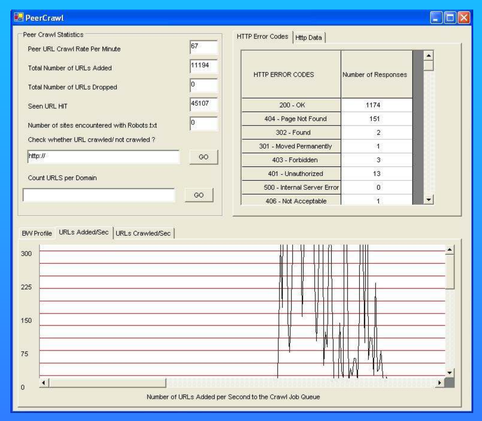
STATS VISUALIZER

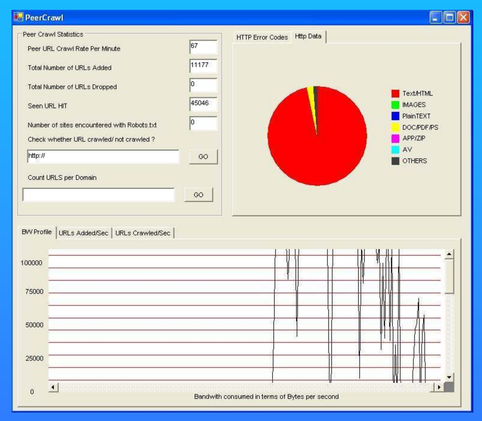
- ❑ Displays the statistics of the crawler in real-time.
- ❑ Single and easily-configurable interface local to each peer.
- ❑ Allows for simple user queries.
- ❑ Displays & classifies crawling rates, size of content downloaded.

SOURCE RANKING

DSPHERE computes two scores: (1) each source is assigned an importance score based on an analysis of the inter-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-resilient countermeasures into the source-based ranking model to support more robust rankings that are more difficult to manipulate than traditional page-based ranking approaches.





Statistics Visualizer for Crawler

PEOPLE

FACULTY
Prof. Ling Liu

CURRENT STUDENTS
Bhuvan Bamba, Tushar Bansal, Joseph Patrao, Suiyang Li

PAST STUDENTS
James Caverlee, Vaibhav Padliya, Mudhakar Srivatsa, Mahesh Palekar, Aameek Singh

DISL

cercs

Georgia Tech College of Computing
Defining the New Face of Computing