Representing Web Graphs

S. Raghaven & H. Garcia-Molina
Computer Science Dept., Stanford University

Amr El-Helw
CS856
University of Waterloo
aelhelw@cs.uwaterloo.ca

Outline

- Motivation
- Contribution
- Introduction
- S-Node Representation
- Experimental Results
- Summary

Motivation

- Efficient traversal of huge Web graphs is a challenging problem.
- The lack of a schema to describe the structure of Web graphs.
- Naive graph representation schemes can increase query execution time.

Contribution

- Proposing a new representation for Web graphs, the "S-Node" representation.
- Demonstrating that S-Node representations are highly space-efficient.
- Showing, by experiment, that S-Node representations can significantly reduce query execution times.

Introduction

- Web Repositories:
  - Large special-purpose collections of Web pages and associated indexes.
- Examples:
  - Research repositories (e.g. Stanford WebBase, the Internet Archive)
  - Commercial search engines (e.g. Google, Altavista)

Introduction

Access to Web Repositories

	Commercial Search Engines	**Research Repositories**
Target Audience	Non-expert users	Expert users
Type of Access	■ Access is controlled by a public search interface. ■ No internal interface (API) is publicly available.	■ Perform complex analysis, mining and indexing over huge data sets. ■ Provide “Bulk” access interface to their content

Introduction

There are kinds of analysis for which both either access mode is unsuitable. They have the following features:

- **Focused Access**
  - It focuses on a small set of pages and associated links (in contrast to a typical mining or analysis task using bulk access).
- **Complex Expressive Queries**
  - It uses predicates on several different properties of pages (e.g. domain, text content), and navigational operations (e.g. pages pointing to other pages).

Introduction

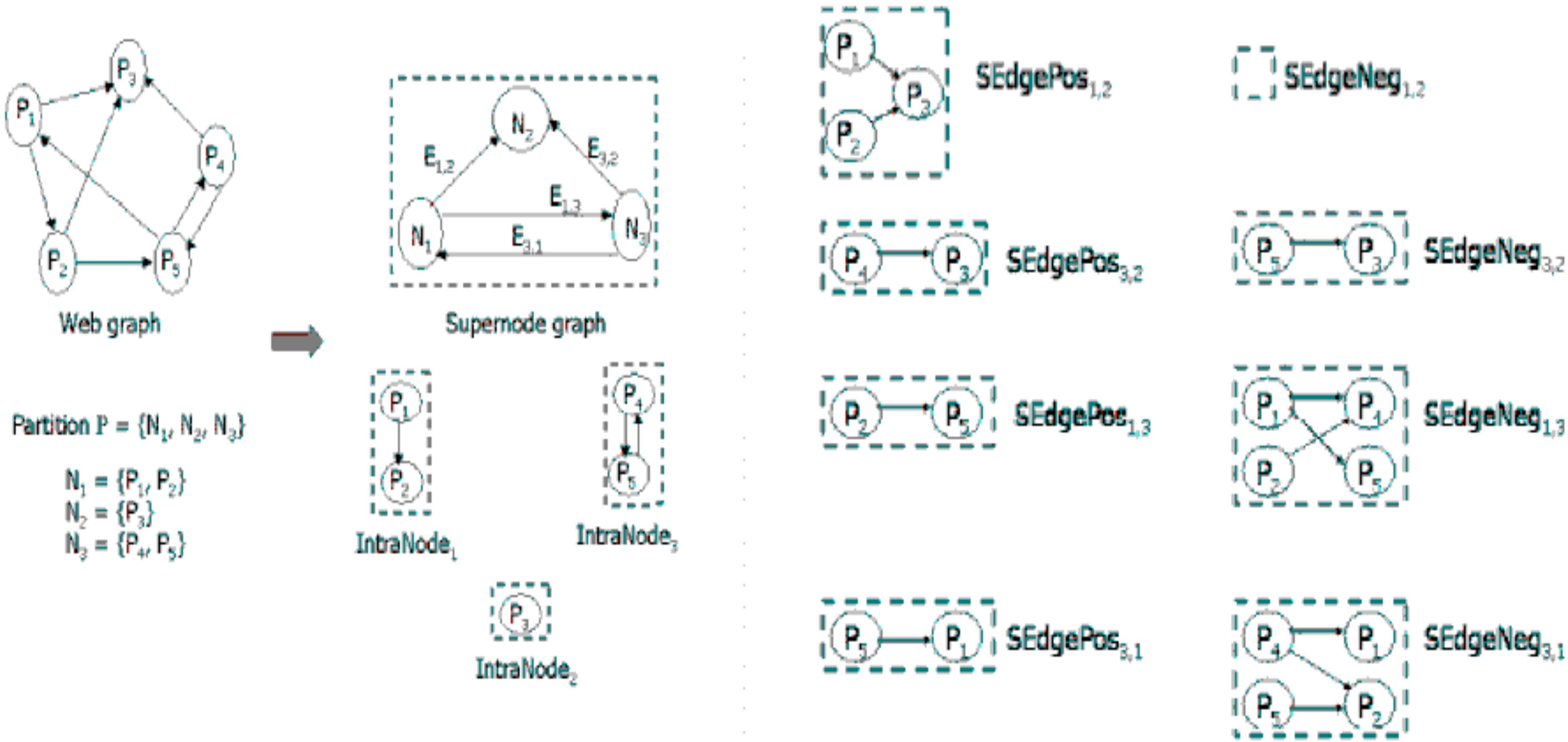
This kind of analysis provides 3 views of the repository:

- A collection of text documents that can be searched and ranked using keywords and/or phrases.
- A navigable directed graph.
- A set of relational tables storing properties (rank, title, domain, …) on which selection, projection and predicates can be applied.

S-Node Representation

- W_G represents the directed Web graph
- Let $P = \{N_1, N_2, \ldots, N_n\}$ be a partition on the nodes (pages) of W_G.
- Some terms of S-Node representation:
  - Supernode graph
  - Intranode graph
  - Positive superedge graph
  - Negative superedge graph

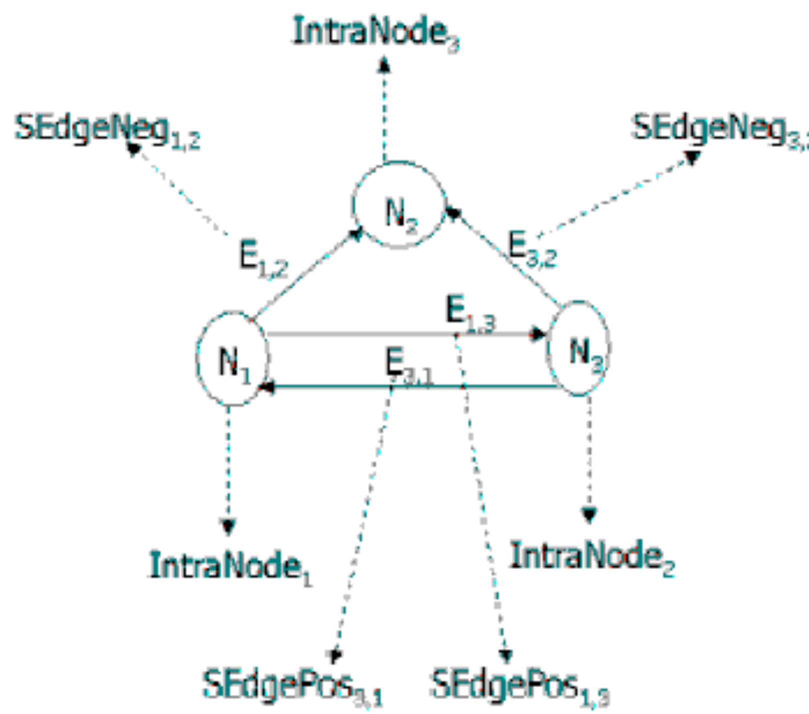
S-Node Representation



S-Node Representation

An S-Node representation of W_G, SNode(W_G, P) can be constructed using all of the following:

- A supernode graph
- A set of intranode graphs
- A set of positive and negative subedge graphs



Building an S-Node Representation

- Requirements for the partition P:
  - It must produce highly compressible intranode and superedge graphs, to achieve a compact representation.
  - For local access queries, the set of pages and links involved must be distributed within a small number of intranode and superedge graphs → Efficient execution by loading only the relevant graphs into main memory.

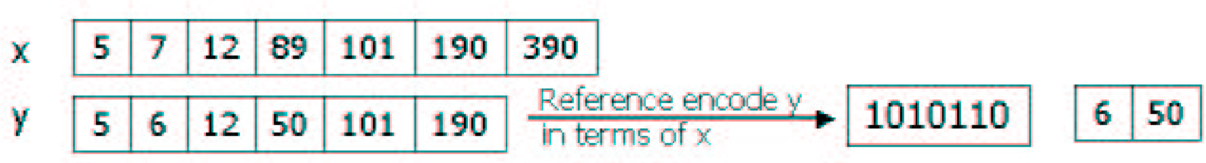
Building an S-Node Representation

- Observations about Web graphs:
  - *Link copying*. There are clusters of pages on the Web that have very similar adjacency lists.
  - *Domain and URL Locality*. Many links from a page point to other pages on the same domain, and possibly with lexicographically close URLs.
  - *Page similarity*. Pages that have very similar adjacency lists (i.e., pages which point to almost the same set of pages) are likely to be related.

Building an S-Node Representation

- Desired partition properties:
  - *Pages with similar adjacency lists are grouped together, as much as possible.*
  - *All the pages assigned to a given element of a partition belong to the same domain.*
  - *Among pages belonging to the same host, those with lexicographically similar URLs are more likely to be grouped together.*

Reference Encoding



- It is a graph compression technique.
- We can compress the adjacency list of *y* by representing it in terms of the adjacency list of *x*
- For each page *x* in a graph *G*, we decide whether the adjacency list for *x* is represented as is or in terms of a reference page, and in that case, the page that will act as reference.
- An affinity graph G_{aff} can be used to encode the intranode and superedge graphs.

Iterative Partition Refinement

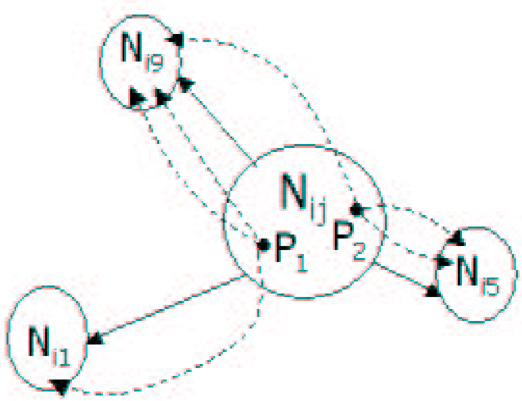
- We begin with an initial coarse-grained partition $P_0 = \{N_{01}, N_{02}, \ldots, N_{0n}\}$.
- This partition is *refined* during successive iterations, generating a sequence of partitions $P_1, P_2, \ldots, P_f$.
- P_0 groups pages based on their domain.
- During every iteration, one of the elements of the existing partition is further broken into smaller pieces.

Iterative Partition Refinement

- **URL Split:**
  - Partitions the pages in N_{ij} based on their URL patterns.
  - Pages that share the same URL prefix are grouped together
  - Every application of URL split on a partition uses a URL prefix, one level/directory longer than the prefix used to generate that partition

Iterative Partition Refinement

- **Clustered Split:**
  - Partitions the pages in N_{ij} by using a clustering algorithm (e.g. k-means), to identify groups of pages with similar adjacency lists.

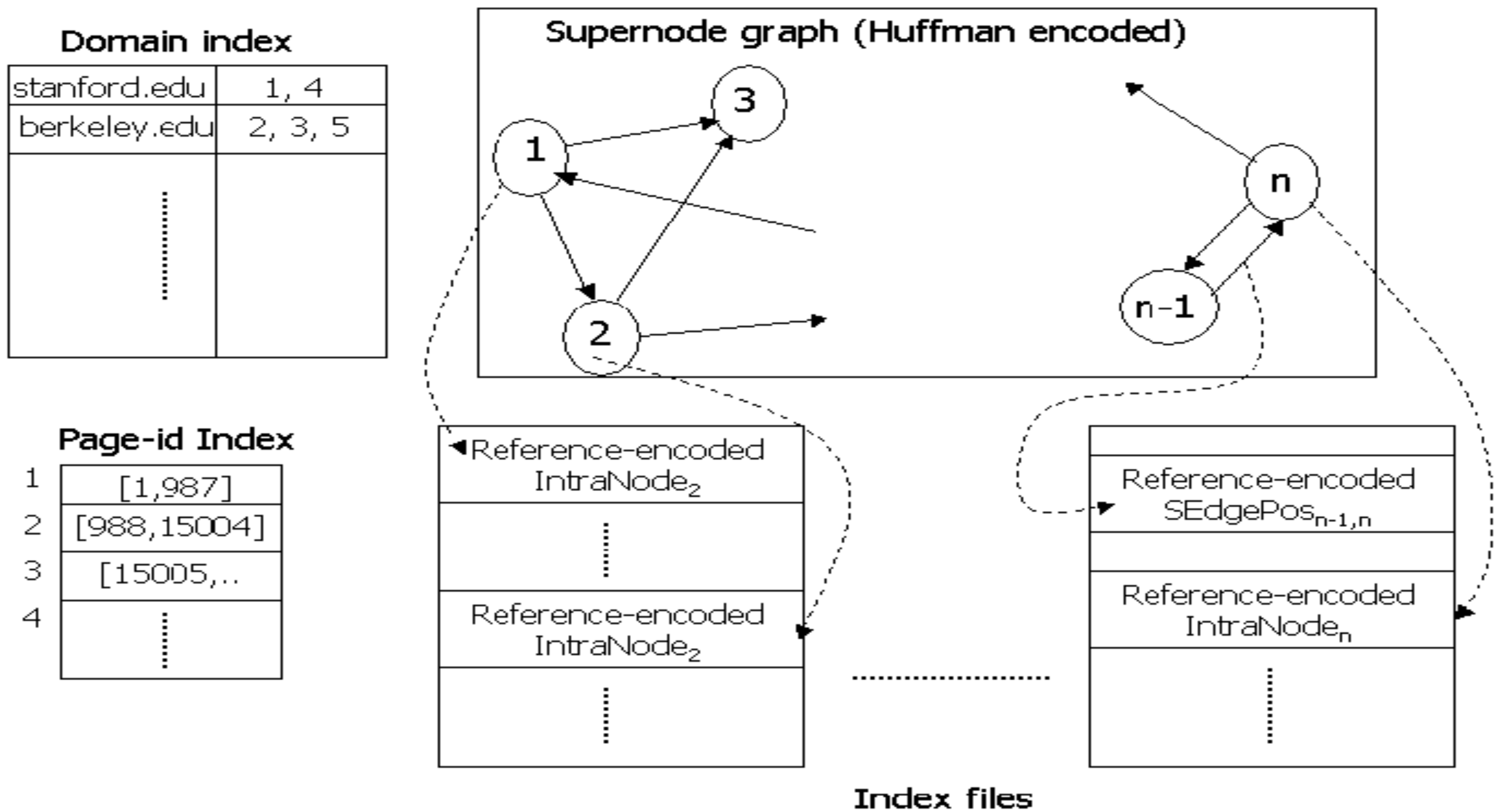


	N_{i1}	N_{i5}	N_{i9}
adj(P_1)	1	0	1
adj(P_2)	0	1	1

Physical Organization

- The supernode graph is encoded using standard adjacency lists.
- A simple Huffman-based compression scheme (based on supernode in-degree)
- Intranode and superedge graphs are encoded using the reference encoding scheme.
- Supernodes are numbered from 1 to n.
- All pages belonging to same supernode are numbered and placed consecutively, in lexicographic order of URLs.

Physical Organization

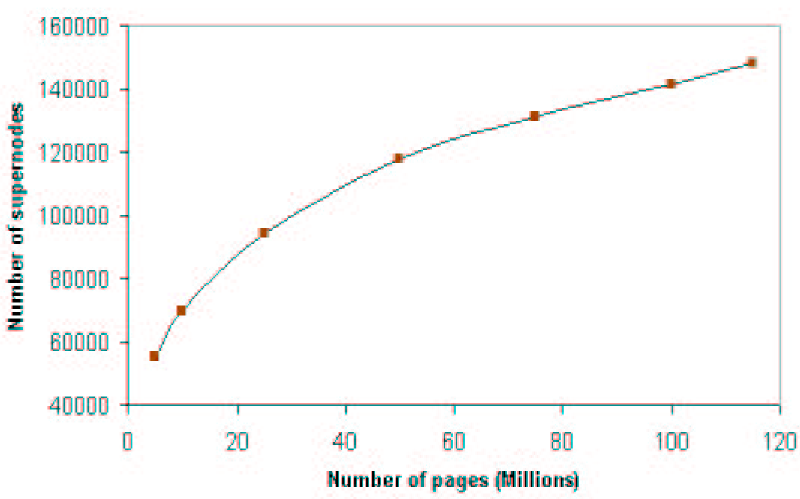


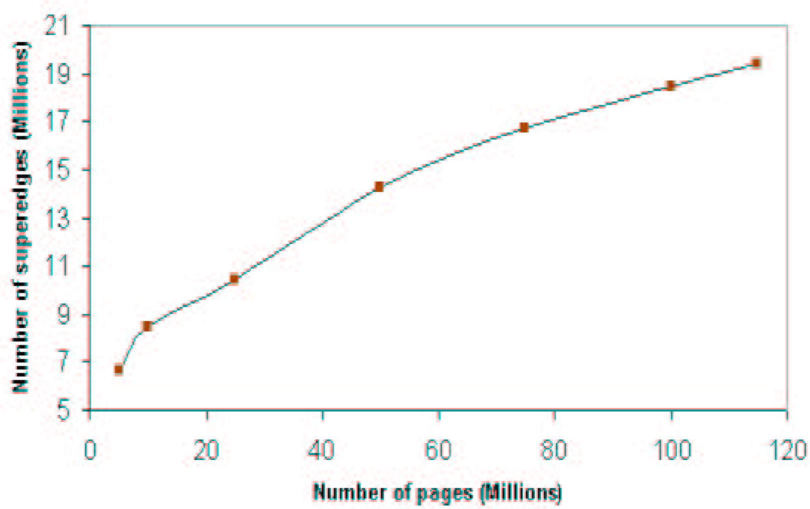
Experimental Results

- **Source data:** about 120 million pages (approximately 900 GB of uncompressed HTML text) from the Stanford WebBase repository, using 5 different-sized data sets.
- The S-Node representation was compared to the following Web graph representation schemes:
  - **Connectivity Server - Link3 scheme**
  - **Huffman-encoded representation** (Huffman codes are assigned to each page based on in-degree).
  - **Relational database. (**using the PostgreSQL object relational database to store the adjacency lists as rows of a database table).
  - **Uncompressed files.**

Experimental Results

- **Scalability Experiments:**
  - From 50 million to 75 million pages (50% increase) → 11% increase in supernodes, and 15% increase in superedges.
  - From 5 million to 100 million pages (20-fold increase) → almost a 3-fold increase in supernodes and superedges.





Experimental Results

- **Compression Experiments:**

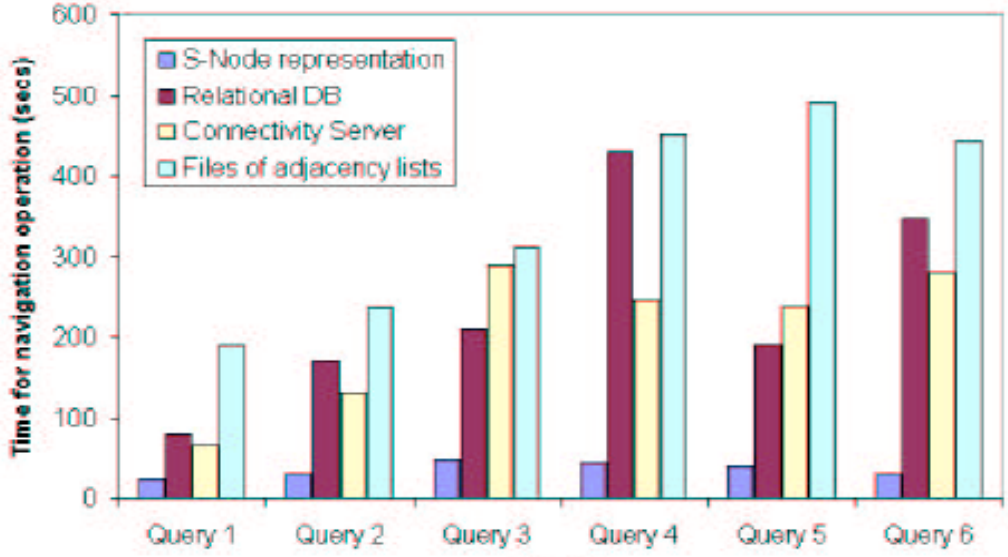
Representation scheme	Number of bits/edge		Max. repository size using 8GB		Seq. access (ns/edge)	Random access (ns/edge)
	W_G	W_G^T	W_G	W_G^T		
Plain Huffman	15.2	15.4	323 million	318 million	112	98
Connectivity Server (Link3)	5.81	5.92	845 million	829 million	309	689
S-Node	5.07	5.63	968 million	872 million	298	702

Experimental Results

- **Complex Queries:** for example:

No.	Description	Main graph operation
1	Generate a list of universities that Stanford researchers working on *Mobile networking* refer to and collaborate with. (Analysis 1 in Section 1).	Subset of the out-neighborhood of a set of pages
2	Compute the relative popularity of three different comic strips among students at Stanford University. (Analysis 2 in Section 1).	Count number of links between 3 different pairs of sets of pages
3	Compute the *Kleinberg base set* [10] for S, where S is the set of top 100 pages (in order of PageRank) that contain the phrase 'Internet censorship'.	Union of out-neighborhood and in-neighborhood of a set of pages
4	Identify the 10 most popular pages on *Quantum cryptography* at each of the following four universities - Stanford, MIT, Caltech, and Berkeley. Popularity of a page is measured by the number of incoming links from pages located outside the domain to which the page belongs.	In-neighborhood for four different sets of pages
5	Suppose S is the set of pages in the repository that contain the phrase *Computer music synthesis*. Rank each page in S by the number of incoming links from other pages in S. Output the top ranked 10 *.edu* pages in S.	Computation of graph induced by a set of pages
6	Suppose $S1$ is the set of Stanford pages (i.e., pages in stanford.edu) that contain the phrase *Optical Interferometry* and $S2$ is the set of Berkeley pages (i.e., pages in berkeley.edu) that contain the same phrase. Let R be the set of pages (not in stanford.edu and berkeley.edu) that are pointed to by at least one page in $S1$ and one page in $S2$. Rank each page in R by the number of incoming links from $S1$ and $S2$ and output R in descending order by rank.	Intersection of out-neighborhoods of two different sets of pages

Experimental Results



Query	Navigation time reduction by using S-Node
1	73.5%
2	76.9%
3	77.7%
4	82.2%
5	79.2%
6	89.2%

Summary

- The paper addresses the problem of efficiently representing massive Web graphs.
- It proposes a novel two-level representation of Web graphs, called an S-Node representation.
- It is based on partitioning the set of pages in the repository.
- S-Node representation can provide impressive compression characteristics (just over 5 bits per edge to represent Web graphs).
- It can also achieve a significant reduction in query execution time (10 to 15 times faster than other schemes for representing Web graphs).