The Information Discovery Graph

Nelson Tang

September 19, 2000

Overview

- The Information Discovery Graph (IDG)
  - Scalable, robust, distributed information directory
  - Rendezvous data sources with clients
  - Low search time, low network overhead

The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

Background

- Amount of online information growing
- Locating relevant information difficult
- Current existing ad-hoc tools
  - Focus on specific niche
  - Low precision (low relevance)
  - Poor scalability

Goals for Good Information Discovery Tool

- Search or browse semantic content
- Handle short- and long-lived information resources
- Low search time
- Low network usage
- Robust
- Scalable

The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

Slide 7 of 70

Previous & Current Work

- Harvest
- Web search engines
- SAP and sdr
- Peer-based data sharing
- DNS

Slide 8 of 70

Harvest

- Early effort at information discovery
- Replicated system of indexed content
- Designed for relatively static content, not dynamic data
- Centrally built index database

Slide 9 of 70

Web Search Engines

- Central database of web pages
- URLs from manual registration, crawling
- Difficulty in relevance (page ranking)
  - Keyword counting
  - Hyperlink analysis
  - Manual evaluation

Slide 10 of 70

Web Search Engines, cont.

- Problems:
  - Central database inherently not scalable
  - Poor crawler coverage
  - Stale entries

Slide 11 of 70

SAP and sdr

- Locates MBone multicast sessions
- Sessions announce to single channel
  - Soft state announcement of all sessions
  - Info pushed to clients, no intermediaries
- Controlling overhead
  - Pre-configured bandwidth limit
  - Announcement frequency determined by number of total sessions
- Wait long enough, will hear everything

Slide 12 of 70

SAP and sdr, cont.

- Problems:
  - Long startup time
  - Unbounded location time if many announcements are missed
  - Searching only at local receiver
    - More overhead at receiver: memory, computation
  - Not suitable for short-lived sessions

Peer-based Data Sharing

- Peer-to-peer networked file system
- Napster: centralized search, registration
- Gnutella: arbitrary mesh, node-dependent searching
- Freenet: no search capability

DNS

- Distributed database
  - Provides mapping between host name and address
- Robustness and scalability
  - Caching and replication
- Designed for relatively static information
- Lookup engine only
  - No relevance rankings

The Information Discovery Graph (IDG)

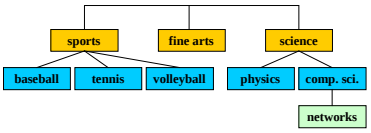
- Addresses information discovery goals
- History:
  - Initially part of Semantic Multicast project
  - Two designs
    - Original design: reduce search time
    - New design: reduce network overhead

The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

Original IDG Design

- Hierarchical taxonomy
  - Semantic tree
  - Higher = broad topic, lower = specific
  - Use structure to speed up queries

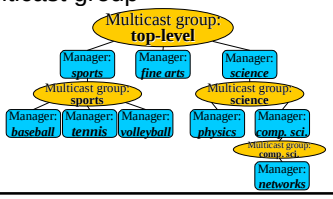


IDG Design, cont.

- Components:
  - Client: users searching for content
  - Data sources: register the content
  - Manager: makes up the directory infrastructure
    - Organization of hierarchy
    - Registration of data sources
    - Handling queries from clients

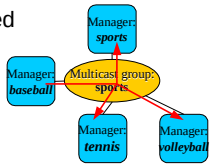
Semantic Hierarchy

- A manager is responsible for a specific topic
- Related managers organized into a multicast group



Inter-Manager Messages

- Periodic messages between managers:
  - Semantic topic
  - Summary of registered data sources
- Purpose:
  - Share information for handling queries
  - Heartbeat

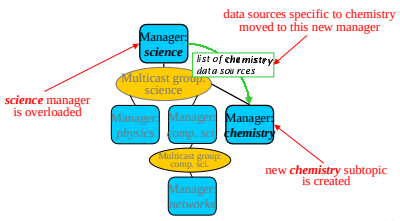


Data Source Registration

- Query for matching manager
- Register, receive acknowledgement
- Soft-state registration
  - Periodically re-register
  - Avoid stale entries
  - Robustness

Load-Balancing

- Adapt to topic popularity



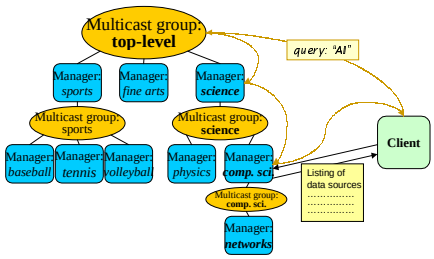
Load-Balancing, cont.

- If manager load above a threshold:
  - Split topic, assign to manager from pool of free managers
  - Pass matching data sources to new manager
- Similarly if manager load below a threshold:
  - Merge the load with parent manager
  - Return the manager to the pool
- IDG hierarchy is fluid

Querying the IDG

- Clients: find the manager with the topic of interest
- Data sources: find the manager to register with
- Top-down search:
  - Start at top
  - Move downward until leaf or match

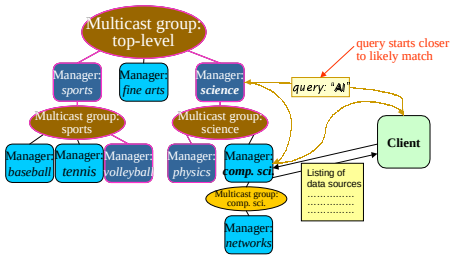
Querying Example



Caching

- As parts of hierarchy are learned, cache them
  - Upper levels relatively stable, easily cacheable
- Start searches lower in tree, closer to match
  - Keep queries out of top-most group
  - Reduce search time
- Minimize caching overhead
  - "Best-effort" consistency
  - Fall back to starting from the top if the cached structure changed

Caching Example



Robustness

- If a manager fails:
  - Other managers miss heartbeat
  - Another manager takes over failed manager's topic
    - Topic is known from previous heartbeat
  - Data sources will notice and re-register
    - During periodic re-registration, will notice manager is gone
    - Randomized re-query to find new manager

Analysis of Original Design

- Compared against SAP
- Two metrics studied:
  - Search time
  - Multicast bandwidth overhead

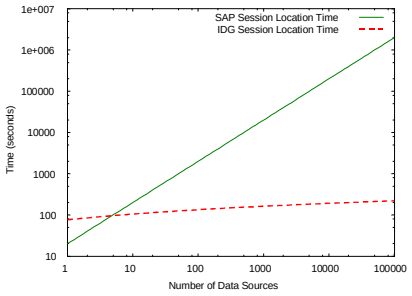
SAP Search Time

- SAP: time to hear all announcements
  N = number of announcements
  interval = max(300,N**ad_size*)/*limit*
- ST_{sap} = N**interval*
- linear in number of data sources

IDG Search Time

- IDG: time to find manager and get list
  N = number of total data sources
  B = branching factor (# of mgrs per group)
  D = depth of tree = $\log_B(N)$
  K = # of data sources per manager
- ST_{idg} = B*D + K
- logarithmic in number of data sources

Search Time Comparison



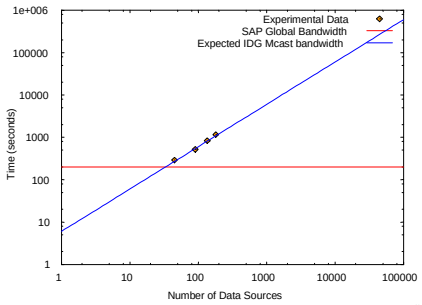
SAP Multicast Bandwidth Usage

- Single global multicast channel
- Pre-configured bandwidth limit
  - SAPv0: default 200 bits/sec
  - SAPv2: default 4000 bits/sec
- constant bound

IDG Multicast Bandwidth Usage

- Many global multicast groups
  N = number of total data sources
  K = # of data sources per manager
  M = number of managers = N/K
  B = branching factor
  L = per-group bandwidth limit
- BW_{idg} = (M/B)*L
- linear in number of data sources

Bandwidth Usage Comparison



Performance Summary

- Search time improved
- Too much multicast bandwidth overhead
- Motivation for new IDG design:
  - Reduce global multicast overhead
  - Retain good search time

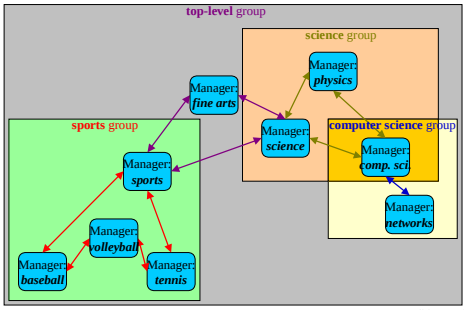
The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

New IDG Design

- Associate location with semantics
  - Each topic has a location
  - Hierarchy has both topic and location info
- Multicast can now be scoped

New IDG Hierarchy Example



Benefits of New Design

- Multicast messages are now limited to within local area only
  - Local heartbeats instead of global
  - Permits scoped access control
- Data sources configured as global or local scope

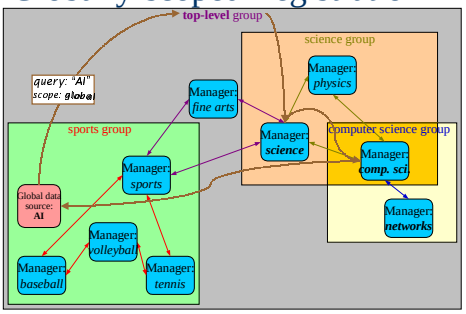
Design Issues of New Model

- Data source registration
- Searching the IDG
- Location of semantic subgroups
- Migration

Data Source Registration

- Globally-scoped data sources:
  - Ignore location
  - Register with semantic match
  - Use top-down search (same as original IDG design)

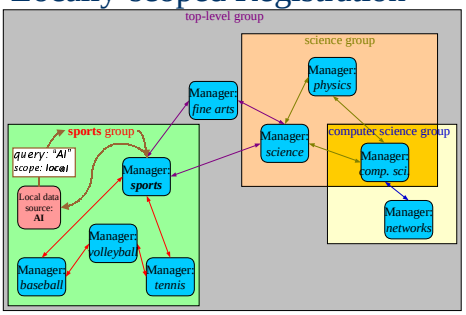
Globally-scoped Registration



Data Source Registration, cont.

- Locally-scoped data sources:
  - Ignore topic
  - Register with smallest scope enclosing the data source

Locally-scoped Registration



Querying the New IDG

- Top-down
- Bottom-up

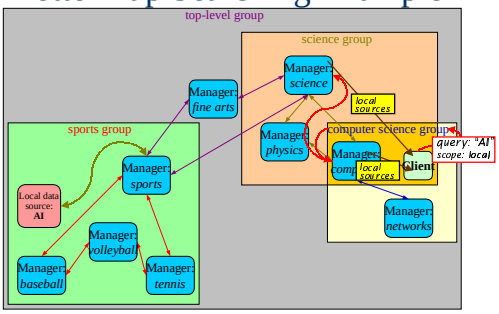
Top-down Searching

- Identical to original IDG design
- Ignores location
- Finds globally accessible data sources

Bottom-up Searching

- Query starts at client's scope, moves upwards towards root
- Ignores semantic topic
- At each step, finds locally-scoped data sources that match topic
- Enables scoped-based access control

Bottom-up Searching Example



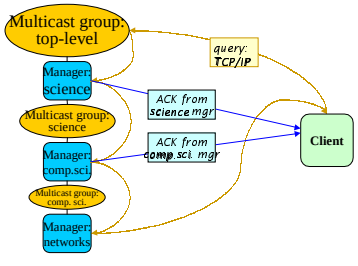
Hybrid Approach

- Use top-down to get globally accessible information
- Use bottom-up to get locally scope-controlled data
- So, use both together to find both kinds of info for a topic

Forwarding Queries

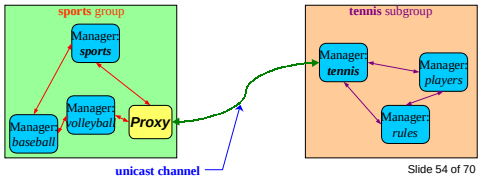
- A query may take several steps to reach the manager(s) with the desired data
- Existing approaches
  - Forward: resend query, no acknowledgement to client (like IP Routing)
  - Redirect: respond to client with next step, client sends query to next hop (like DNS)
- IDG approach: when a manager gets a query:
  - forward query to next node
  - send acknowledgement to the client
- Helps client discover IDG hierarchy

Forwarding Queries Example



Location of Semantic Subgroups

- Child groups can be far away from the parent manager
- Maintain scoped multicast with proxy
  - Child manager multicasts to group via proxy



Proxy Message Handling

- Proxy acts as relay between remote manager and local multicast group
- Heartbeats:
  - Aggregate and forward to manager
- Queries:
  - Proxy responds for the manager
  - Queries do not need to be forwarded

Proxy Failure Recovery

- Proxy responsible if attached manager fails
  - Signaled by loss of unicast connection
  - Request new manager from remote group's pool of free managers
  - Establish unicast connection with newly assigned manager
- Similarly, if proxy fails, attached manager requests a new proxy

Migration

- If many group members are proxies:
  - Move location of group to make managers local again
  - Side effect: move towards "hot spot" of data source popularity
  - Use free manager pool at destination

Analysis

- Search time
- Multicast bandwidth overhead

Search Time

- Top-down part: identical
  - logarithmic in number of data sources
- Bottom-up part: time to walk path to root
  - logarithmic in number of data sources
- Total search time: sum of top-down and bottom-up parts
- logarithmic in number of data sources

System Overhead

- Scalability affected by *global* multicast, not by *local* multicast
  - Only top-most group managers are globally distributed
  - So, only heartbeats among top-most managers contribute to global message overhead
- constant bound

New IDG Performance Summary

- Search time kept to logarithmic
- Global multicast reduced from linear to constant

Other Sources of Overhead

- Examine other sources of overhead
  - Data source registration overhead
  - Proxy overhead
  - Migration overhead
- Analyze impact on model

The Plan

- Background
- Previous & Current Work
- Original IDG Design
- New IDG Design
- Current Status & Future Work

Current Status & Future Work

- Now: implementing Parsec simulation
- Future:
  - Verify overhead estimates
  - Implement prototype
  - Anticipated issues
    - Differentiate scope and locality
    - Semantic organization of taxonomy
    - Deployment

Differentiating Scope and Locality

- Scope not always matched to physical locality
  - e.g., administrative scoping
- Can different kinds of scoping be used?

Semantic Organization of Taxonomy

- Information is not simple hierarchy
  - Many cross-references
  - Complex taxonomy
- Mesh structure more accurate
  - Multiple parents, multiple children
  - Fanout of queries

Deployment Issues

- Anticipate issues of deploying IDG
  - Who deploys managers for manager pool?
  - Who decides where topics will be assigned?

Slide 67 of 70

Manager Deployment

- Motivation to deploy IDG managers
  - Increase chance that popular topic is local
  - Reduce response time for local users
- Top-level managers may need governmental agency backing
  - Minimum deployment
  - Similar to DNS root server deployment

Slide 68 of 70

Topic/Location Pairings

- How will topics and locations be initially paired?
  - Doesn't matter, rely on adaptive migration
  - Ensure migration scheme is efficient

Slide 69 of 70

Conclusion

- IDG provides information discovery
  - Search or browse by semantic categories
  - Low search time, low network overhead
  - Robust
  - Scalable
- Goal: develop and deploy IDG as the standard information discovery tool

Slide 70 of 70