Supporting Rapid Processing and Interactive Map-Based Exploration of Streaming News

Hanan Samet*

hjs@cs.umd.edu

Department of Computer Science

University of Maryland

College Park, MD 20742, USA

*Based on joint work with Michael D. Lieberman

NewsStand: Spatio-Textual Aggregation of News and Display

1. Crawls the web looking for news sources and feeds
   - Indexing 10,000 news sources
   - About 50,000 news articles per day
   - 300GB of data
2. Aggregate news articles by both content similarity and location
   - Articles about the same event are grouped into clusters
3. Rank clusters by importance which is based on:
   - Number of articles in cluster
   - Number of unique newspapers in cluster
   - Event's rate of propagation to other newspapers
4. Associate each cluster with its geographic focus or foci
5. Display each cluster at the positions of the geographic foci
6. Other options:
   - Category (e.g., General, Business, SciTech, Entertainment, Health, Sports)
   - Image and video galleries
   - User-generated news (e.g., Social networks such as Twitter)
   - Map stories by people, disease, etc.

NewsStand's Capabilities

- Advantage: map, coupled with ability to pan and vary zoom level at which it is viewed, provides an inherent granularity to the search process that facilitates an approximate search
- Distinguished from today's prevalent keyword-based conventional search methods that provide a very limited facility for approximate searches
  - Realized by permitting a match via use of a subset of keywords
  - Users have no idea as to which keyword to use, and thus would welcome a capability for search to also take synonyms into account
  - Map query interface is a step in this direction
  - Act of pointing at a location (e.g., by the appropriate positioning of a pointing device) and making the interpretation of the precision of this positioning specification dependent on the zoom level is equivalent to permitting the use of spatial synonyms
  - Ex: Seek a "Rock Concert in Manhattan"
  - "Rock Concerts" in "Harlem" or "New York City" are good answers when none in Manhattan" as correspond to spatial synonyms: "Harlem" by proximity, and "New York City" by containing Manhattan

Motivation: Change News Reading Paradigm

- Use a map to read news for all media (e.g., text, photos, videos)
- Choose place of interest and find topics/articles relevant to it
- Topics/articles determined by location and level of zoom
- No predetermined boundaries on sources of articles
- Application: monitoring hot spots
  1. Investors
  2. National security
  3. Disease monitoring
- One-stop shopping for spatially-oriented news reading
  1. Summarize the news
     - What are the top stories happening?
  2. Explore the news
     - What is happening in Darfur?
  3. Discover patterns in the news
     - How are the Olympics and Darfur related?
- Overall goal: make the map the medium for presenting all information with a spatial component

Existing News Readers

1. Usually only have a rudimentary understanding of the implicit geographic content of news articles, usually based on the address of the publishing news source (e.g., newspaper)
   - Usually present articles grouped by keyword or topic; not by geography
2. Microsoft Bing
   - Rather primitive and top stories presented linearly
   - Little or no classification by topic
3. Google News Reader
   - Classifies articles by topic
   - Local news search
     - Aggregates articles by zip code or city, state specification
       - E.g., articles mentioning “College Park, MD”
     - Provides a limited number of articles (9 at the moment)
     - Seems to be based on the host of the articles
       - E.g., “LA Times” provides local articles for “Los Angeles, CA”
     - Seems to use Google Search with location names as search keys
       - E.g., articles for ZIP 20742 are those mentioning “College Park, MD” or “University of Maryland”
     - Has no notion of story importance in the grand scheme
     - International versions use international news sources

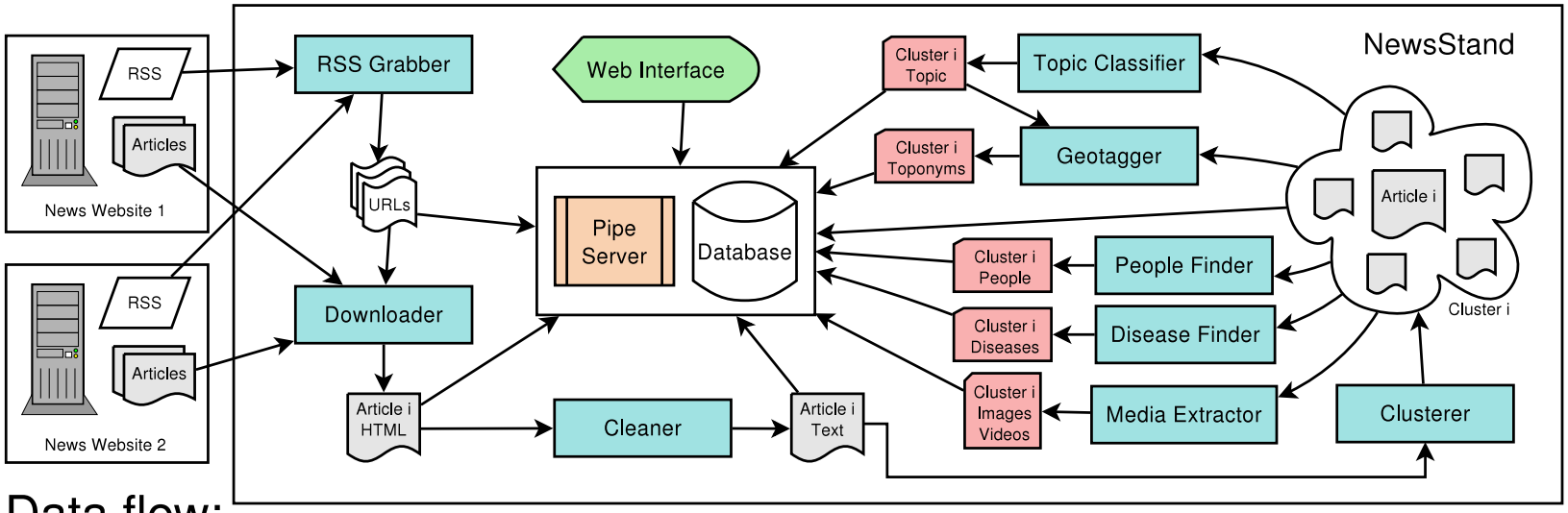
Map Query Interface Requires a Geotagger

- Geotagger: processor that converts a textual specification of a location to a geometric one (i.e., latitude-longitude pair)
- Geotagging issues:
  1. Toponym recognition: identify geographical references in text
     - Does “Jefferson” refer to a person or a geographical location?
  2. Toponym resolution: disambiguate a geographical reference
     - Does “London” mean “London, UK”, “London, Ontario”, or one of 2570 other instances of “London” in our gazetteer?
  3. Determine spatial focus of a document
     - Is “Singapore” relevant to a news article about “Hurricane Katrina”?
     - Not so, if article appeared in “Singapore Strait Times”

Live Demo

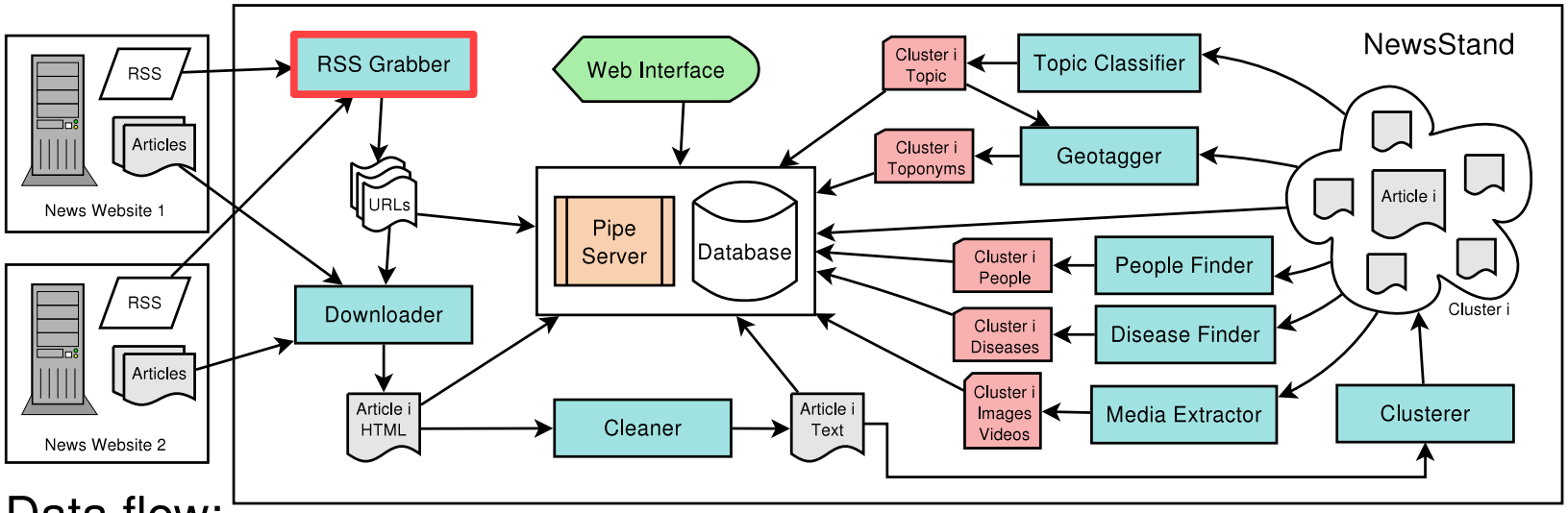
http://newsstand.umiacs.umd.edu
http://newsstand.umiacs.umd.edu/news/mobile

NewsStand's Architecture



Data flow:

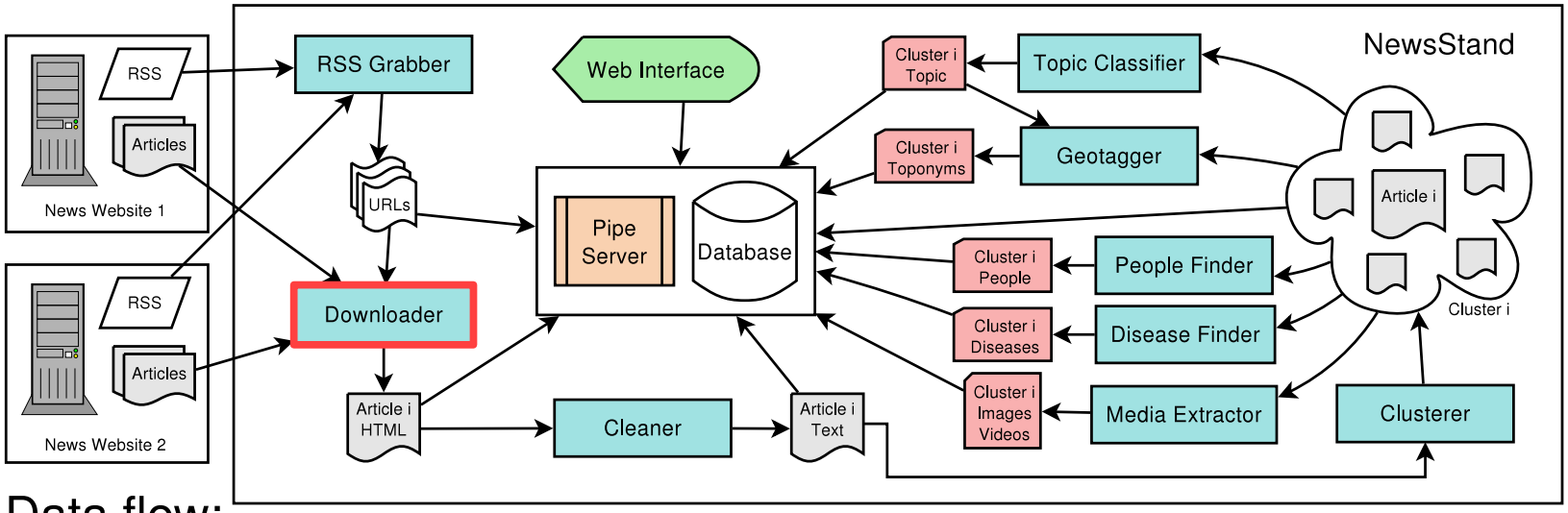
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.

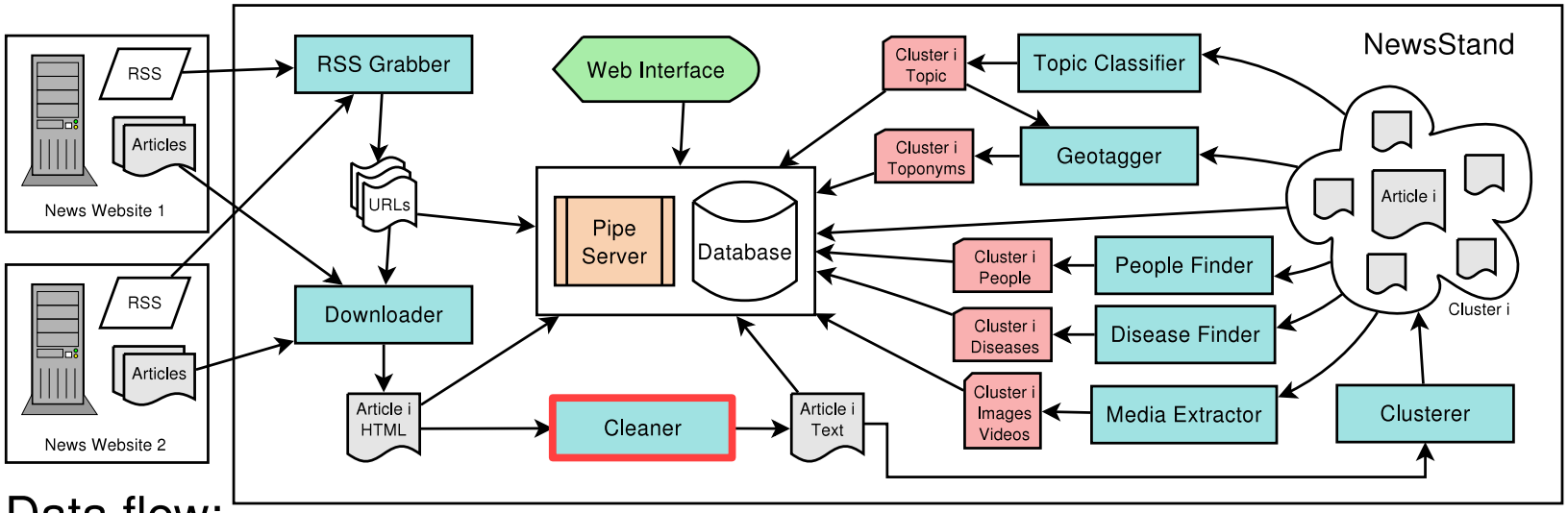
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.

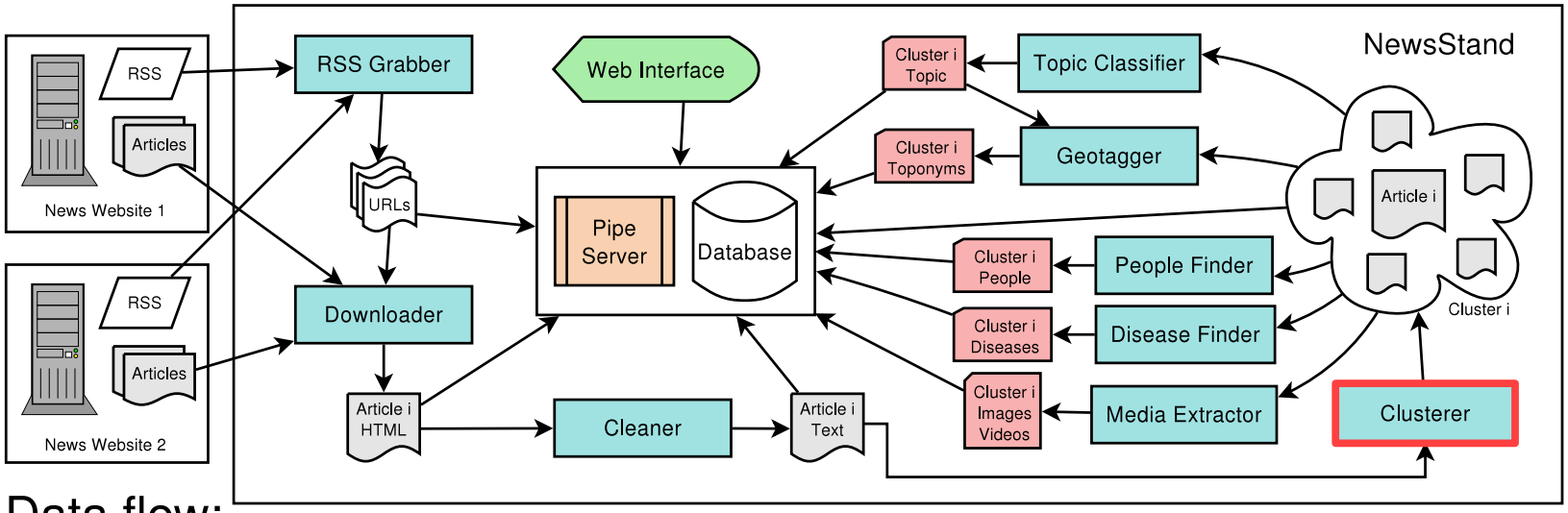
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.

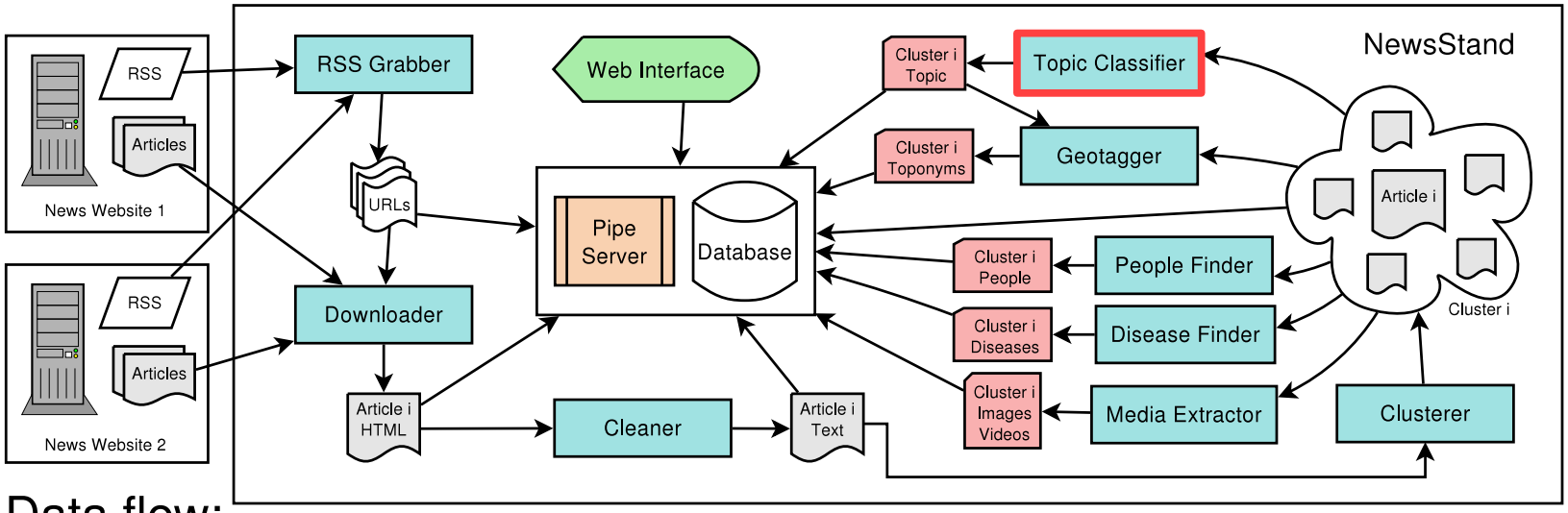
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.

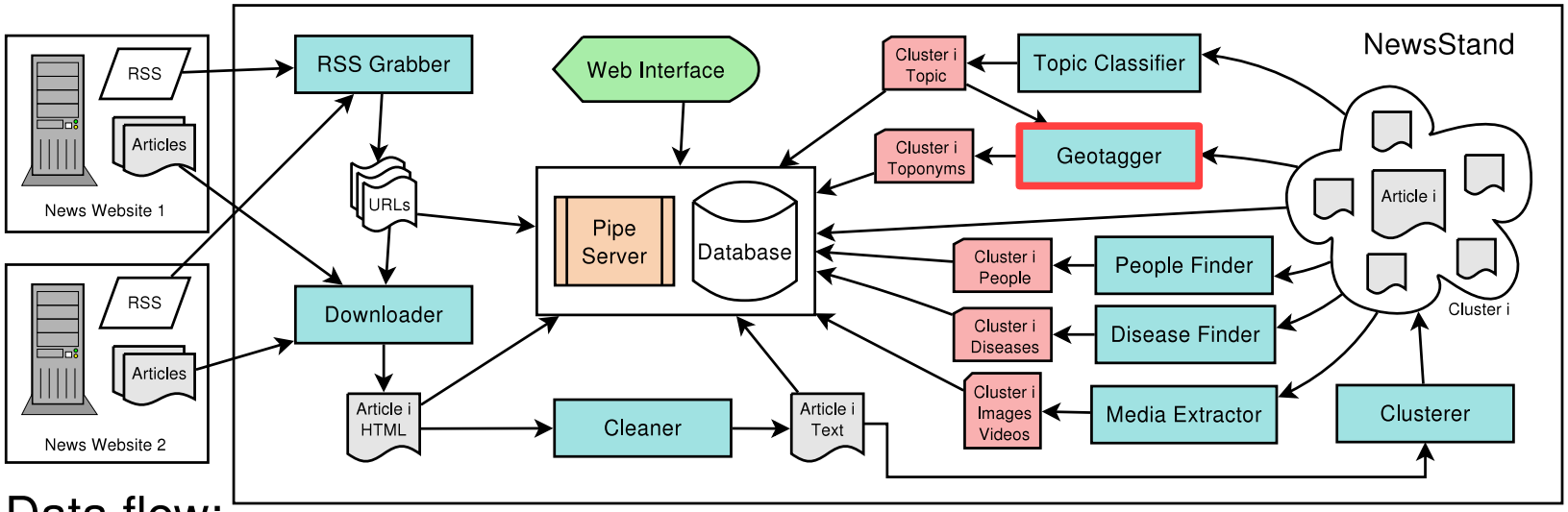
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.
5. **Topic Classifier**: Assigns general topics to articles (e.g., "Sports").

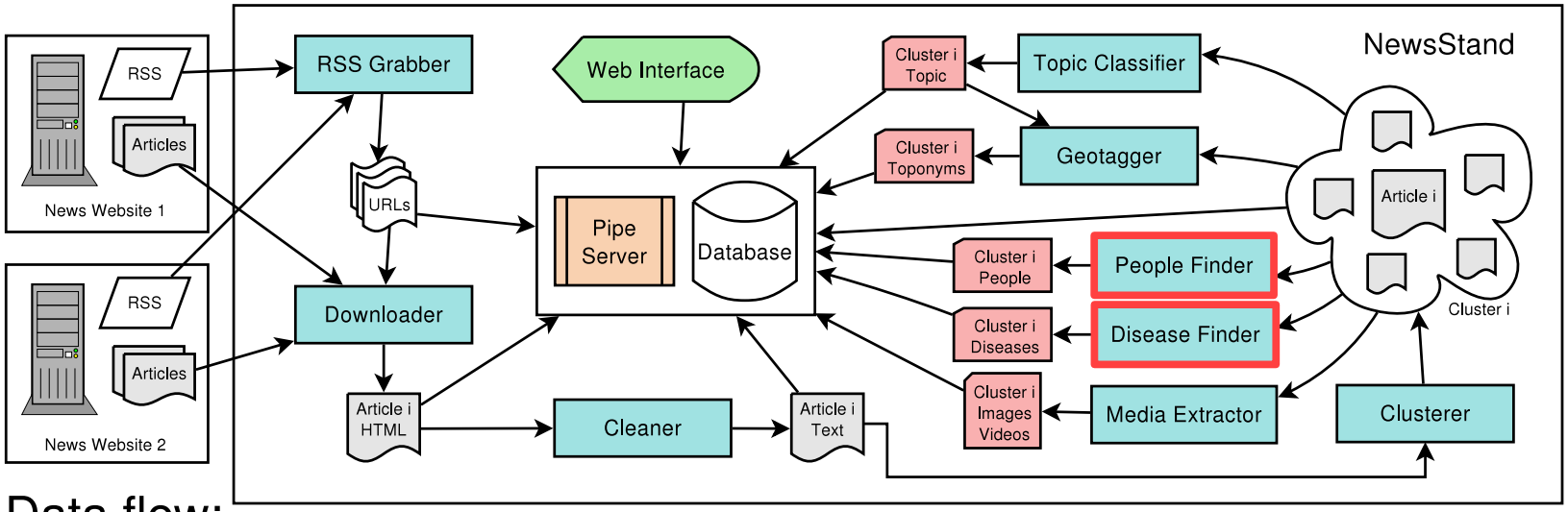
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.
5. **Topic Classifier**: Assigns general topics to articles (e.g., "Sports").
6. **Geotagger**: Finds toponyms and assigns lat/long values to each.

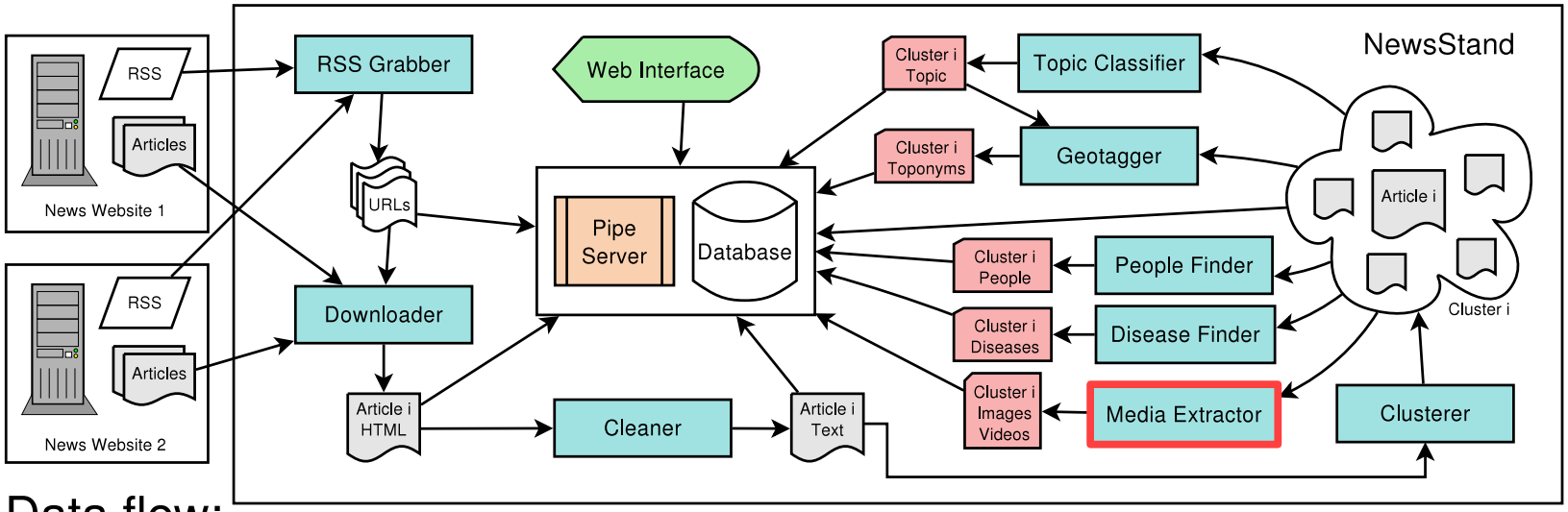
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.
5. **Topic Classifier**: Assigns general topics to articles (e.g., "Sports").
6. **Geotagger**: Finds toponyms and assigns lat/long values to each.
7. **People/Disease Finder**: Finds mentions of people/diseases.

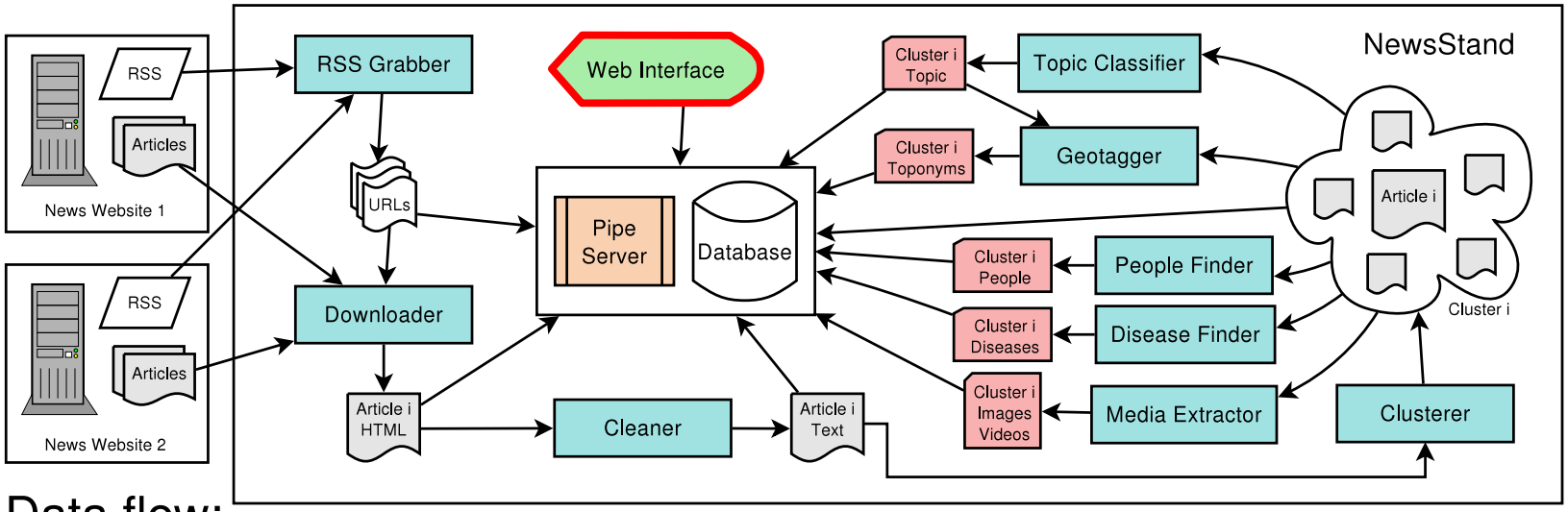
NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.
5. **Topic Classifier**: Assigns general topics to articles (e.g., "Sports").
6. **Geotagger**: Finds toponyms and assigns lat/long values to each.
7. **People/Disease Finder**: Finds mentions of people/diseases.
8. **Media Extractor**: Extracts captioned images and videos.

NewsStand's Architecture



Data flow:

1. **RSS Grabber**: Polls RSS feeds and retrieves URLs to news articles.
2. **Downloader**: Downloads news articles from URLs.
3. **Cleaner**: Extracts article content from source HTML.
4. **Clusterer**: Groups together articles about the same story.
5. **Topic Classifier**: Assigns general topics to articles (e.g., "Sports").
6. **Geotagger**: Finds toponyms and assigns lat/long values to each.
7. **People/Disease Finder**: Finds mentions of people/diseases.
8. **Media Extractor**: Extracts captioned images and videos.
9. **Web Interface**: Accesses database to retrieve data for display.

Architecture

- Backend processing organized as a pipeline
- Documents stream in and flow through different processing stages performed by slave modules connected to the pipeline
- Each module performs a different function with later modules in the pipeline often depending on results of earlier modules
- Challenge lies in coordinating modules so that can execute simultaneously without much idle time
- Means of control and synchronization:
  1. Pipe server tracks documents as they flow through the processing pipeline and assigns documents to slave modules
  2. SQL database stores information about documents and results of each processing stage
- Both the pipe server and the database have their own communication protocols to which modules must adhere
- Note the decoupling of NewsStand's control channel (pipe server) from its data channel (database)
- Retrievals are graphical and translated to SQL queries on the database

Pipe server

- Coordinates backend processing by assigning batches of processing to slave modules via a communication protocol
- Maintains a collection of work queues called *pipes*
- One pipe per slave type which process a collection of document ids (docids)
- Keeps track of performance of slave modules so that it can restart them as well as given them more work when they are done
- Pipes and relevant docids are stored in a disk-based hash which does not depend on NewsStand's database
- When communicating with slave modules, instead of waiting for immediate responses from each slave, which could slow processing, the pipe server employs non-blocking input/output and buffering
- Each connected slave is tracked individually with regard to the work batch sent to it, and this work does not move to the next pipe until the slave sends back a valid work complete response
- Design assume slaves are not malicious
- Drawback of design is that pipe server is a single point of failure
  - If pipe server fails, so does NewsStand
  - Experience that only cause of failure of pipe server was rebooting

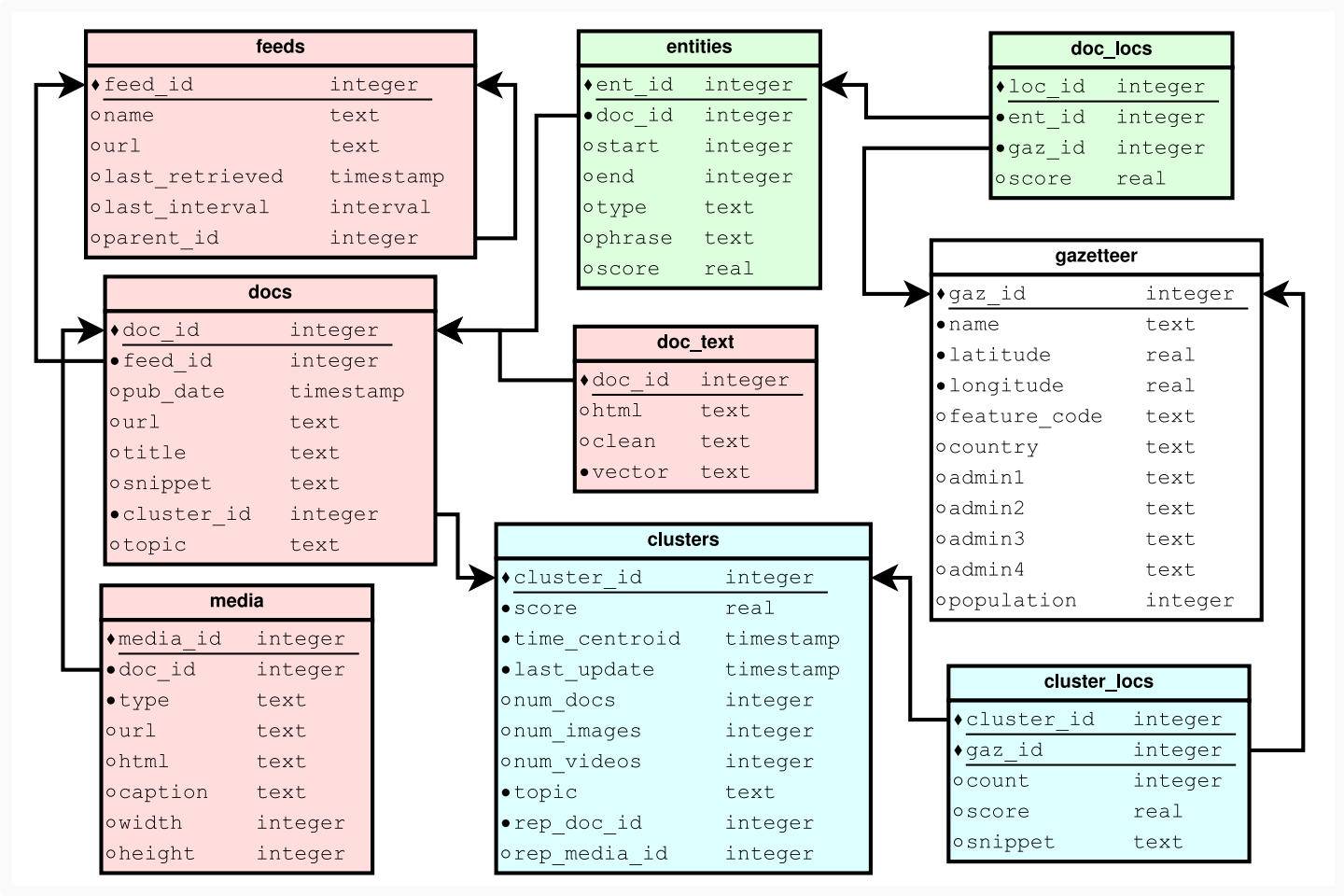
Document Processing Workflow

- Document identifiers are added to the pipe server by the RSS Grabber, the first module
- When slave module starts it connects to both pipe server and database
- Pipe server sends batch of document identifiers to the slave instance
- For each document
  1. Slave instance retrieves information about document from database
  2. Performs its processing on document
  3. Stores results in database
- Upon finishing processing all documents in batch, slave instance
  1. Reports back to the pipe server that batch is finished
  2. Receives another batch of documents to work on
- Individual instances of modules are directed by the pipe server to work on independent batches of documents
  - Avoids processing bottlenecks by starting additional instances of slave modules that require more processing time, allowing greater scalability

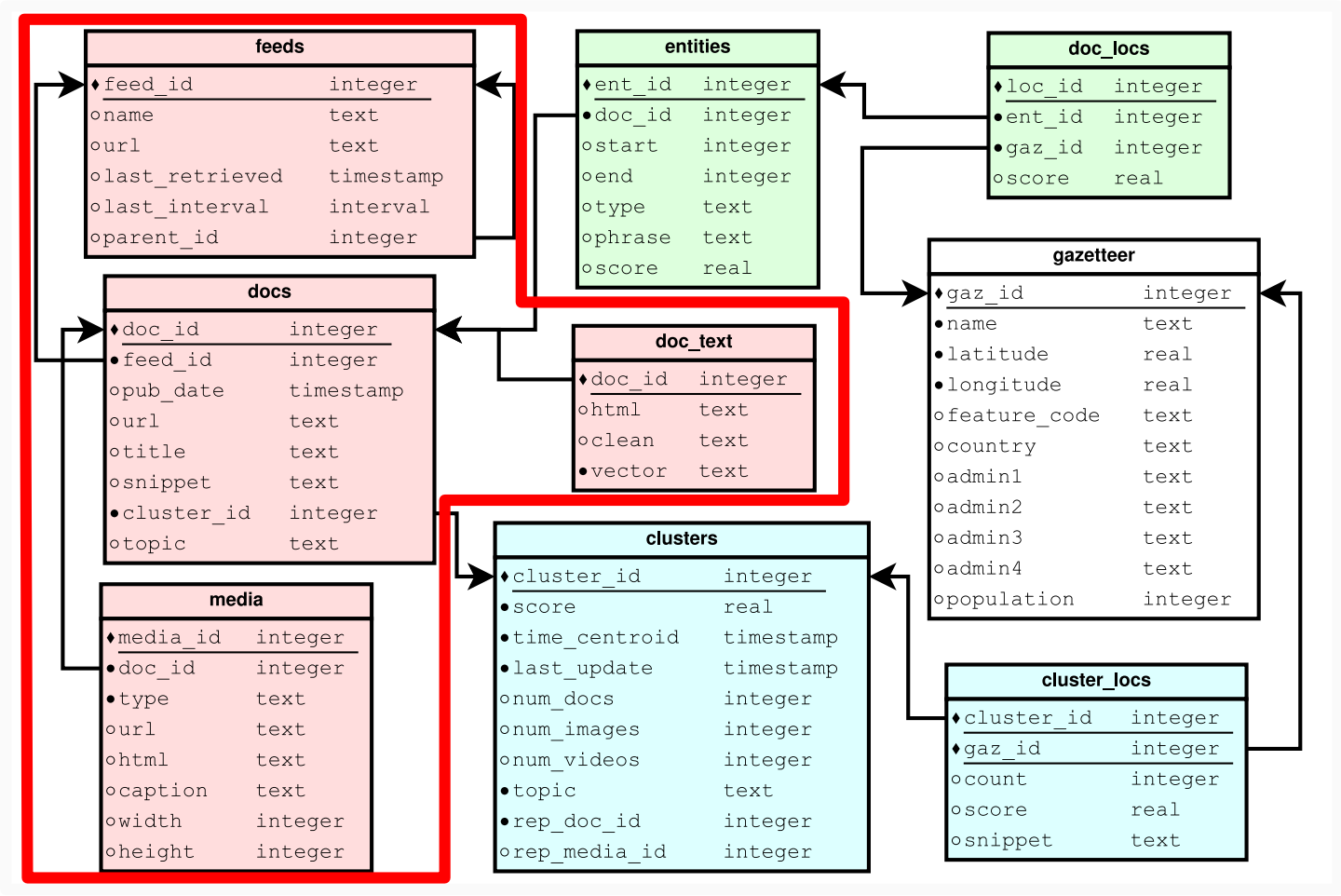
Database Design

- Two goals
  - Managing data streaming through NewsStand due to its processing
  - Serving NewsStand's map query interface to facilitate interactive browsing and exploration of news
- Use PostgreSQL relational database
- Much data churn as data constantly added and deleted from database unlike typical SQL databases where data is mostly static
- Heavy data churn means statistics gathered for query optimizer go out of date quickly and thus must perform vacuuming often
- Maintain a second cache database containing only the recent data in order to improve service to the map query interface

NewsStand Database Schema

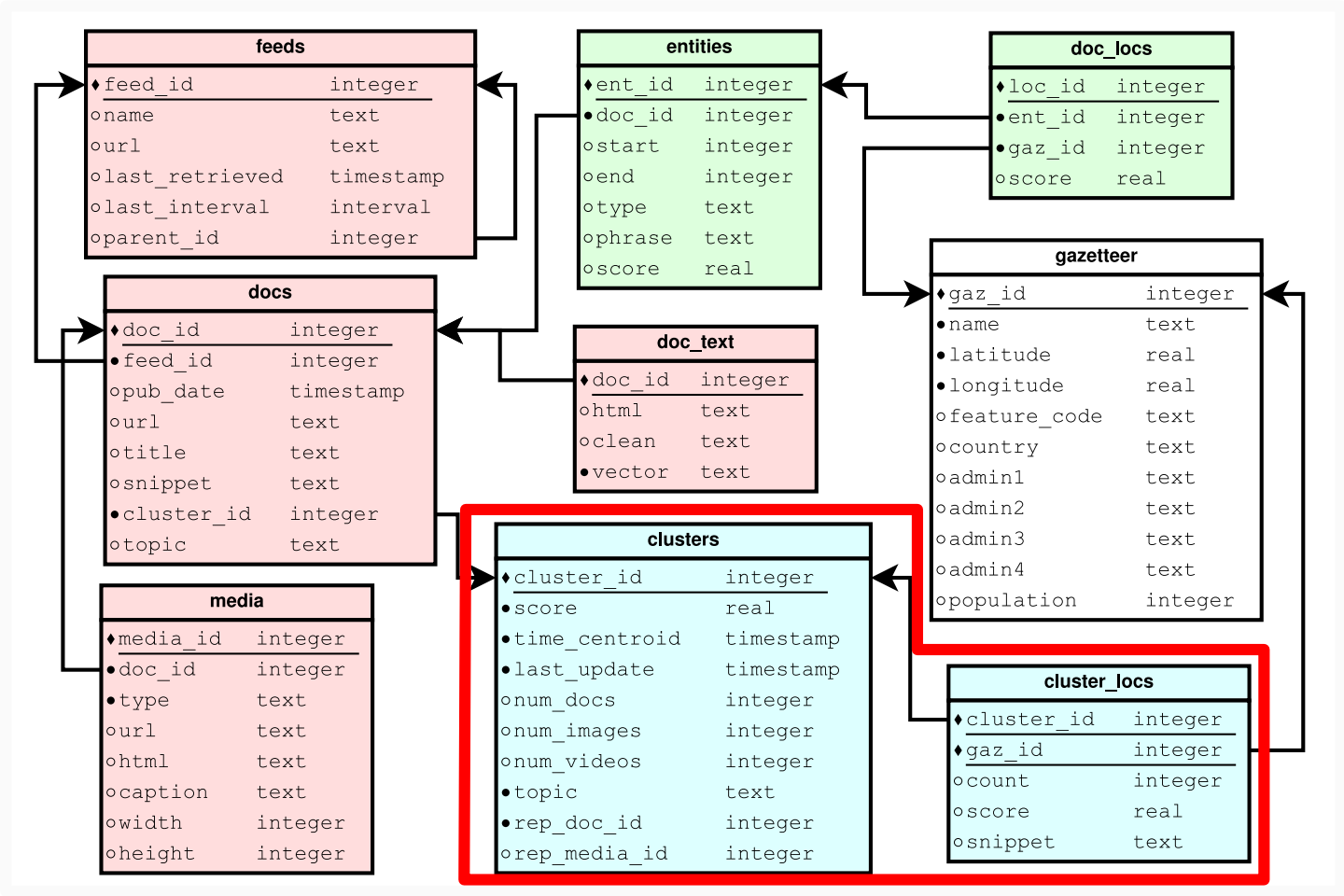


NewsStand Database Schema



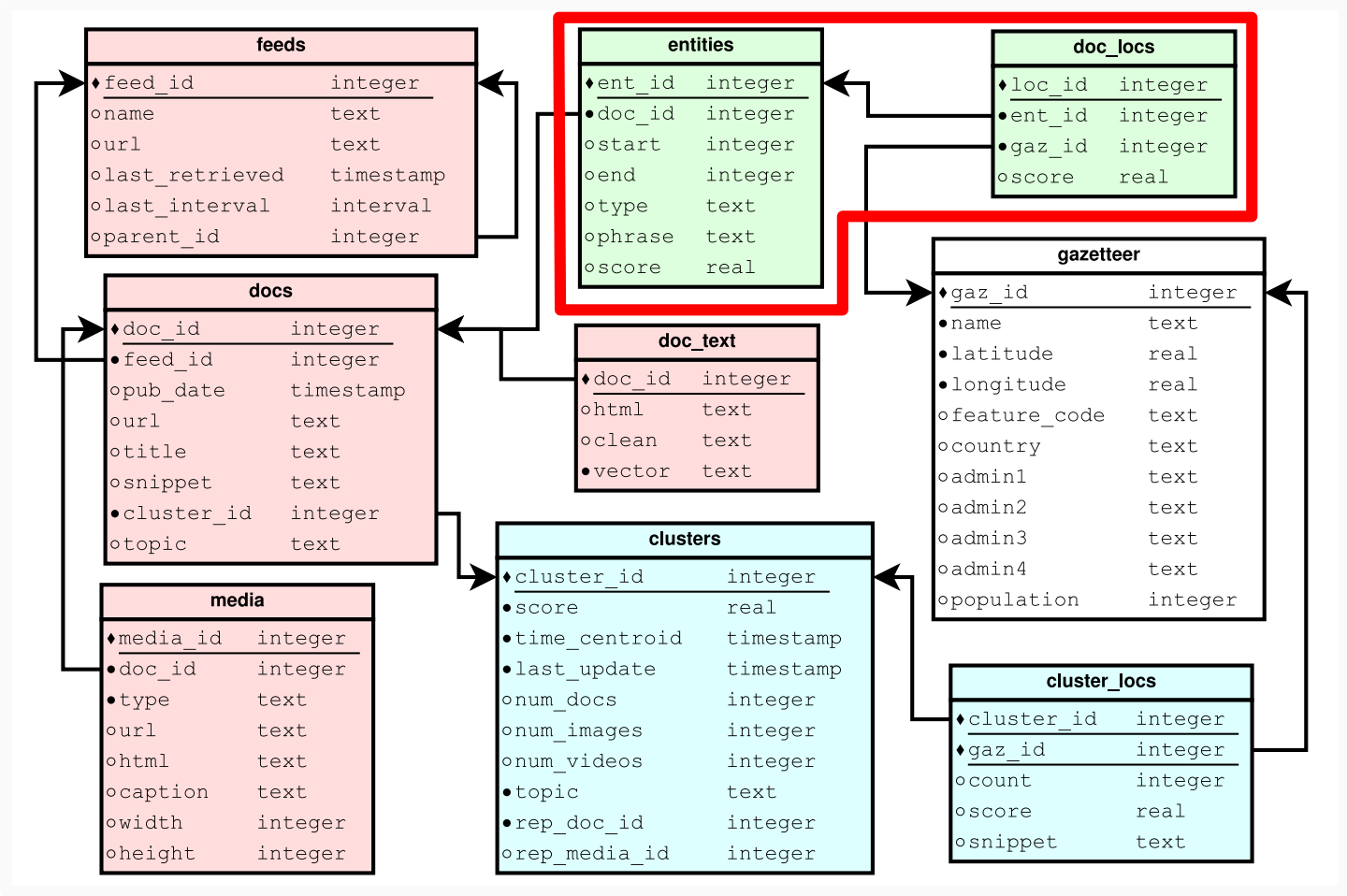
- Document Tables

NewsStand Database Schema



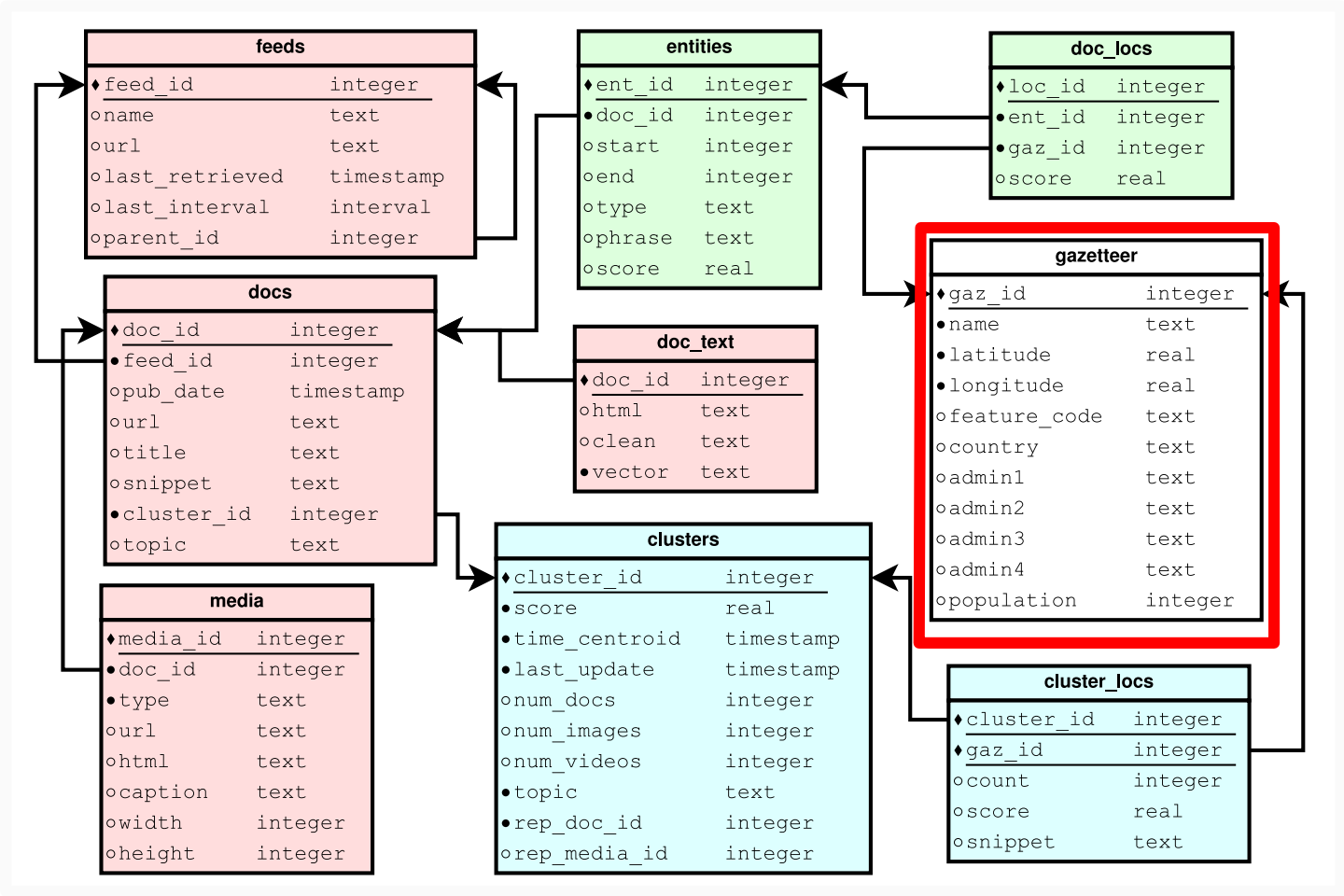
- Document Tables, Cluster Tables

NewsStand Database Schema



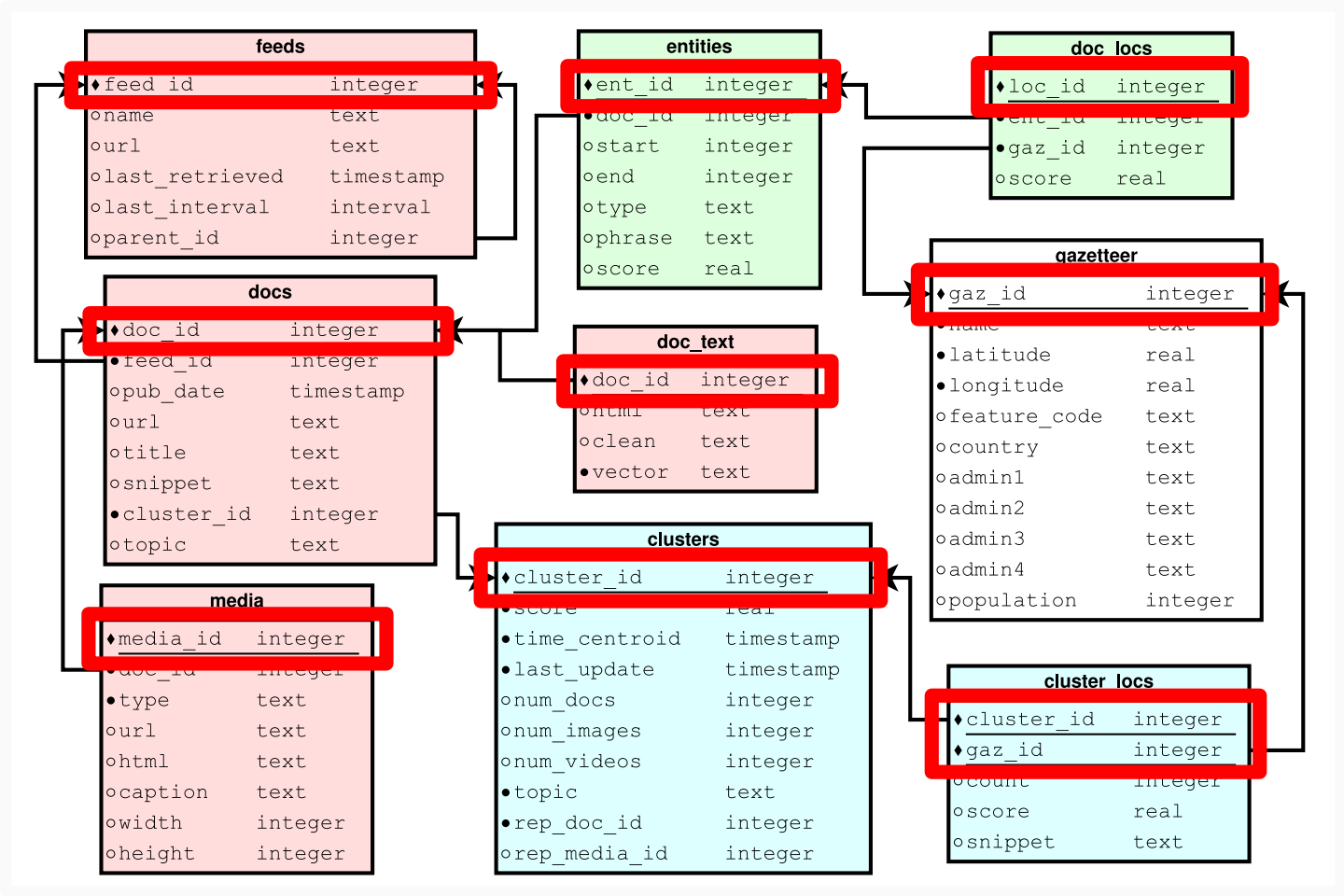
- Document Tables, Cluster Tables, Location Tables

NewsStand Database Schema



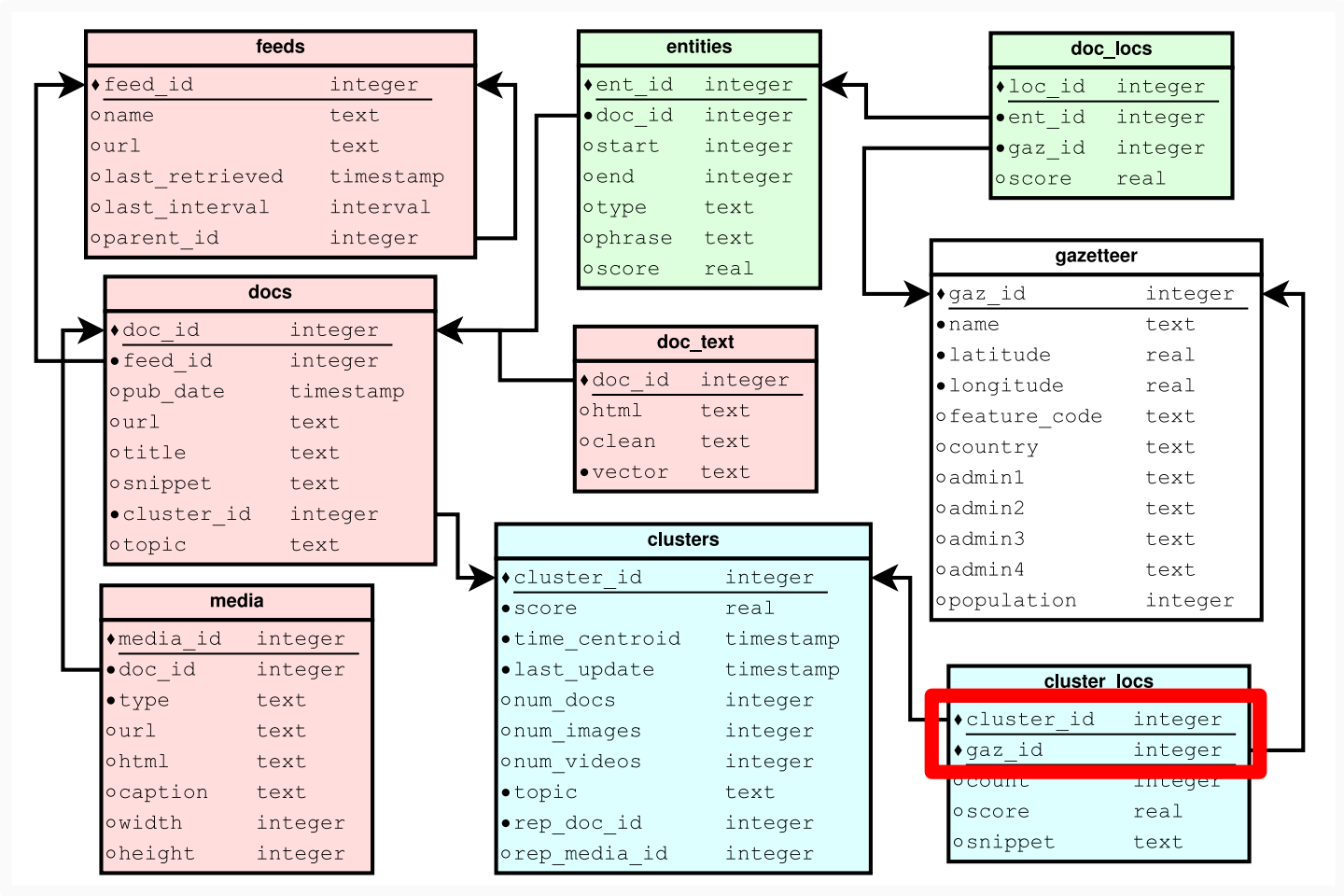
- Document Tables, Cluster Tables, Location Tables, External Data Tables (e.g., Gazetteer, Disease, and People Ontologies)

NewsStand Database Schema



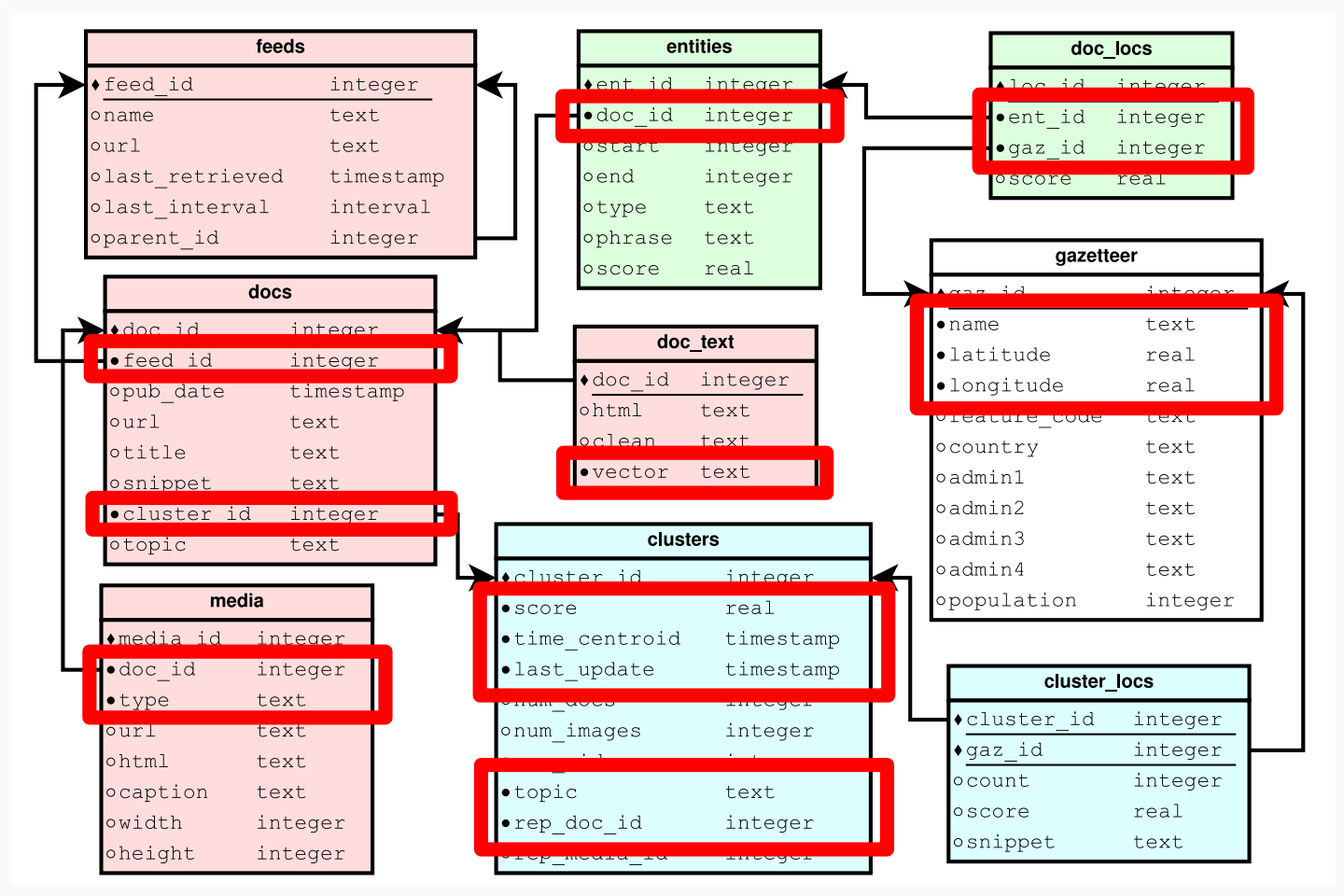
- Document Tables, Cluster Tables, Location Tables, External Data Tables (e.g., Gazetteer, Disease, and People Ontologies)
- Filled diamonds=Primary Keys

NewsStand Database Schema



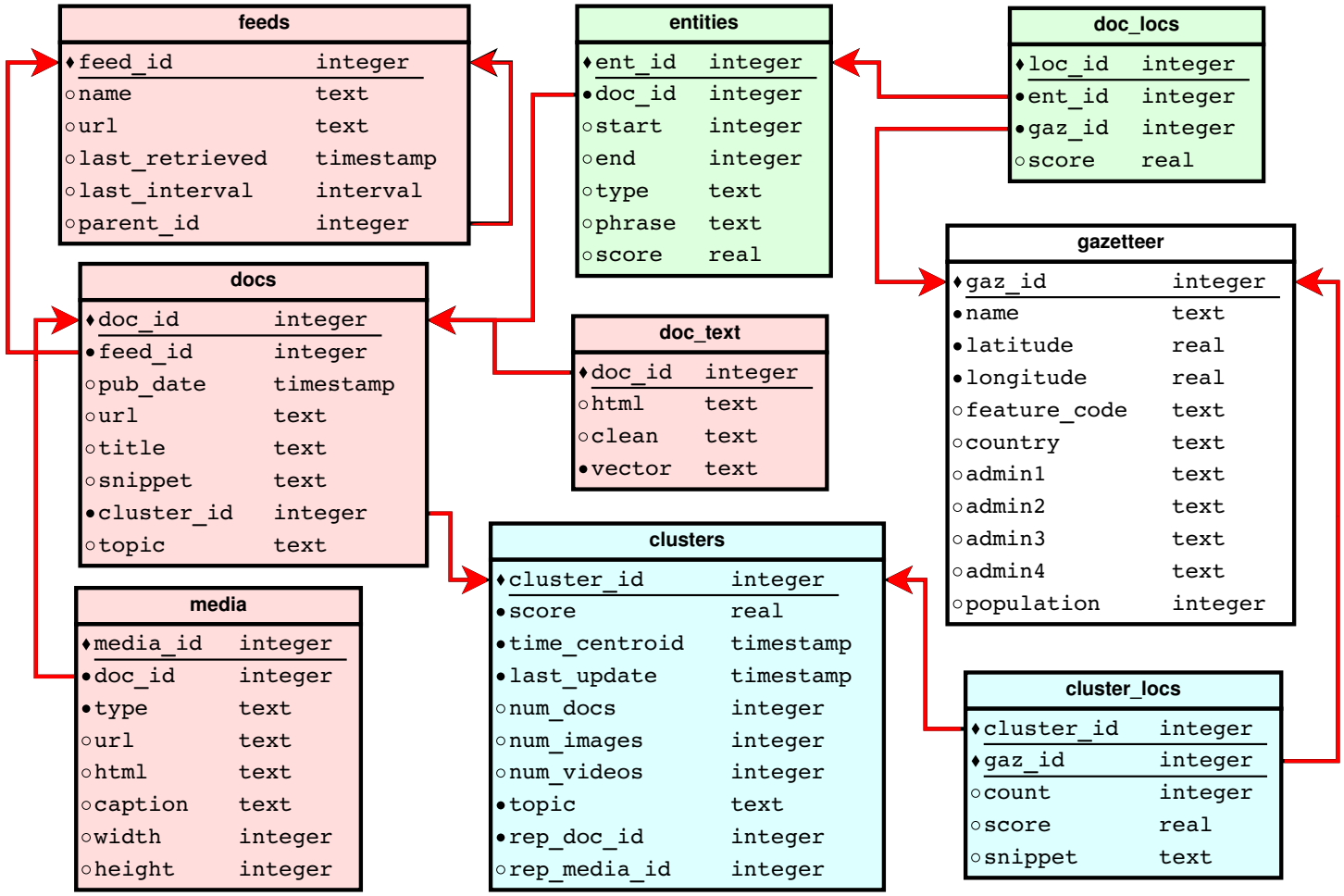
- Document Tables, Cluster Tables, Location Tables, External Data Tables (e.g., Gazetteer, Disease, and People Ontologies)
- Filled diamonds=Primary Keys (multi-column: many-to-many relationship)

NewsStand Database Schema



- Document Tables, Cluster Tables, Location Tables, External Data Tables (e.g., Gazetteer, Disease, and People Ontologies)
- Filled diamonds=Primary Keys (multi-column: many-to-many relationship), Filled circles=Indexed Attributes

NewsStand Database Schema



- Document Tables, Cluster Tables, Location Tables, External Data Tables (e.g., Gazetteer, Disease, and People Ontologies)
- Filled diamonds=Primary Keys (multi-column: many-to-many relationship), Filled circles=Indexed Attributes, Arrows=Foreign Keys

Documents

- RSS Grabber ingests new documents by polling RSS feeds
  - If new data is found, then decrease polling interval
  - Otherwise, increase it
- `parent_id` field stores an optional field used when multiple RSS feeds belong to the same news source
  - E.g., New York Times has a business feed, sports feed, etc.
- `doc_text` contains html, cleaned full text, and the vector space representation

Gazetteer

- Represented by `gazetteer` table containing
  - Latitude/longitude values
  - Metadata
- Based on GeoNames, a crowdsourced gazetteer with over 7.5 million locations
- Updated nightly by synchronizing with GeoNames

Cache Database (DB)

- Main database serves as a data repository (300GB)
  - Downloads 50K articles per day and stores 4 months worth of news
  - Stores full text rather than just the url
  - Constantly in communication with the processing modules with many tuple updates (i.e., inserts, updates, deletes)
    - Could cause some SELECT statements to block while waiting for transactions to complete
    - Means database not suitable to serve NewsStand's user interface (UI)
- Maintain separate cache DB (6.4GB)
  - Only contains most recent news (several days)
  - Same schemas as main database but just smaller tables
  - Means more table rows can be in the database's cache buffers
  - Queries do not modify main database and thus will not block on waiting for modification
  - Cache DB must be updated from main database so serve latest news
    - Special cache updating module continuously polls main database for clusters whose last_update value is newer than previous update time and copies cluster and associated information to cache DB
    - Minimize updates to cache DB so processing resources can serve UI

Database Queries

- Two basic queries
- Top stories mode: Where is topic X happening?
  - Spatial data mining
- Map mode: What is happening at location Y?
- All are variants of top-k queries
  - Return first k ranked results according to some ranking function
- Many of the queries correspond to joins among multiple tables in the database
- Sometimes use materialized views

Top Stories Mode

- Where is topic X happening
  - Slider enables varying number of locations so return top k
- Ex: Get top k locations for cluster *cid*

```
SELECT * FROM clusters c, cluster_locs cl, gazetteer g
WHERE c.cluster_id = cl.cluster_id
   AND cl.gaz_id = g.gaz_id   AND c.cluster_id = cid
ORDER BY cl.score LIMIT k
```

Map Mode

- What is happening at location Y?
  - When Y is viewed as a single location, then query returns all clusters associated with Y
  - Slider, coupled with interpreting Y as a region corresponding to viewable area on display, enables varying number of locations and thereby the number of clusters effectively returning top k clusters
  - Can also devise a ranked spatial range or ranked spatial join query where we need some order in which to deliver clusters

Map Mode Example

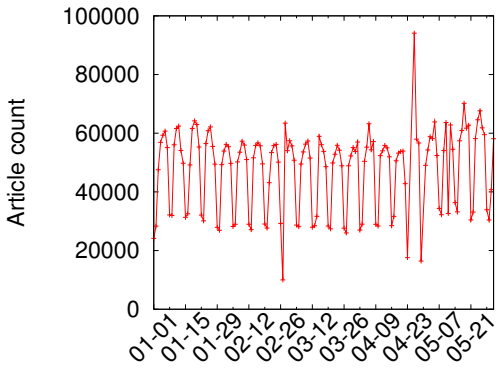
- Ex: Get top k clusters in query window qw

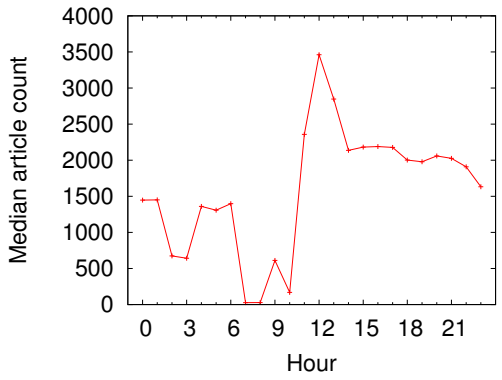
```
SELECT DISTINCT c.cluster_id
FROM clusters c, cluster_locs cl, gazetteer g
WHERE c.cluster_id = cl.cluster_id
   AND cl.gaz_id = g.gaz_id
   AND contains(qw, g.latitude, g.longitude)
ORDER BY c.score LIMIT k
```

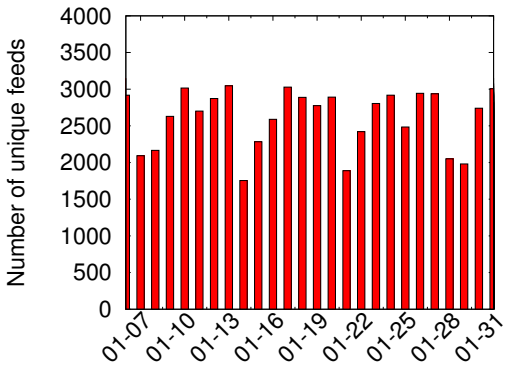
Experimental Results

- Goals
  - Characterize streaming news collected by NewsStand
  - Characterize processing time
  - Characterize querying time
- Different from conventional experiments which use small corpora from well-known sources like New York Times, Reuters, etc.
  - About 50,000 articles per day
  - Dwarfs typical corpus size of hundreds
- Conducted over the live NewsStand database system over several months
- Therefore provide a better view of the long term performance of NewsStand over streaming news

Data Collection

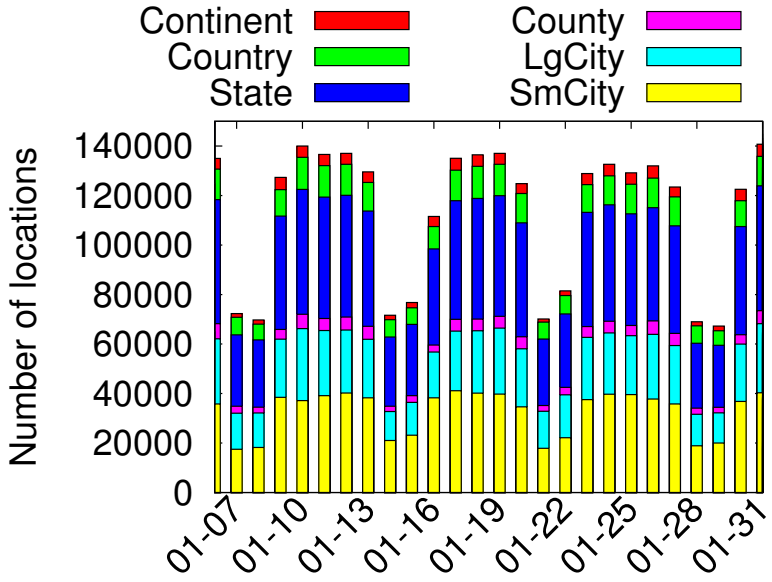






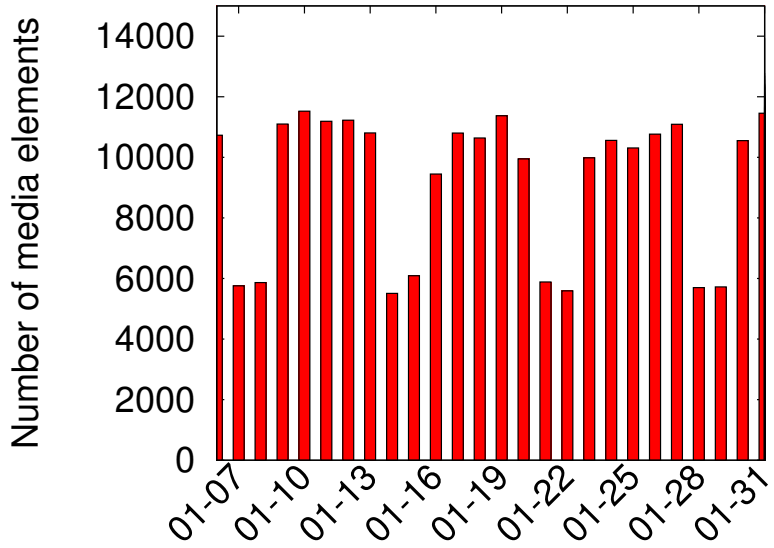
- Measured over a 5 month period from January to May 2011
- Most articles in daytime hours (50,000) rather than weekend (30,000)
- Bias towards US sources with heaviest load between 11:00 and 13:00
- Dips and peaks in article counts on 26 Feb and 25 Apr correspond to system downtime and subsequent catchup recovery
- Total of 10,000 unique feeds and an average of 2,000-3,000 per day

Location Content



- Measure content found by NewsStand's Geotagger
  - Number and types of locations
- Same Weekday-versus-weekend pattern as before
- $\approx$ 130,000 locations on weekdays and $\approx$ 3–4 locations per article
- Smaller places including small cities (under 100,000 population) and stes dominate the location type counts, followed by larger places including large cities (over 100,000 population) and countries
  - Vindicates NewsStand's focus on highly local streaming news
  - Harder to geotag but yields much richer local news experience

Multimedia Content



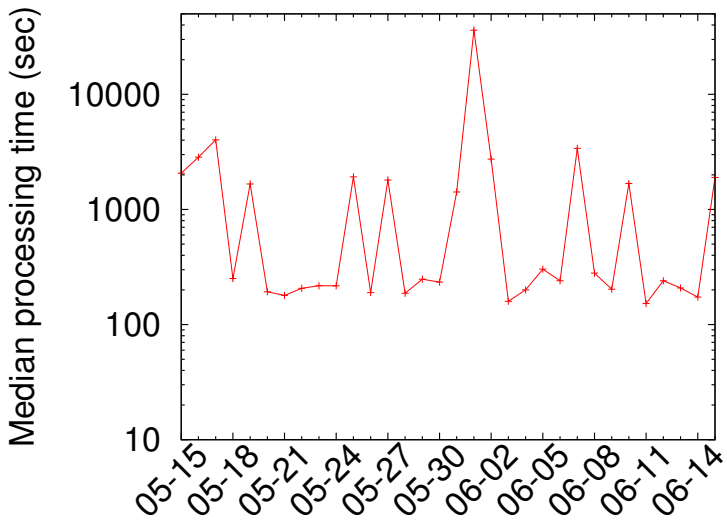
- Measure content found by NewsStand's Media Extractor
  - Amount of multimedia (images and videos)
- Same Weekday-versus-weekend pattern as before
- $\approx$ 11,000 extracted media on weekdays
- Relatively low number of images and videos, $\approx$ 1 image per 3–4 articles
  - Shows difficulty of extracting relevant images and image captions while also filtering for advertising and other spurious media

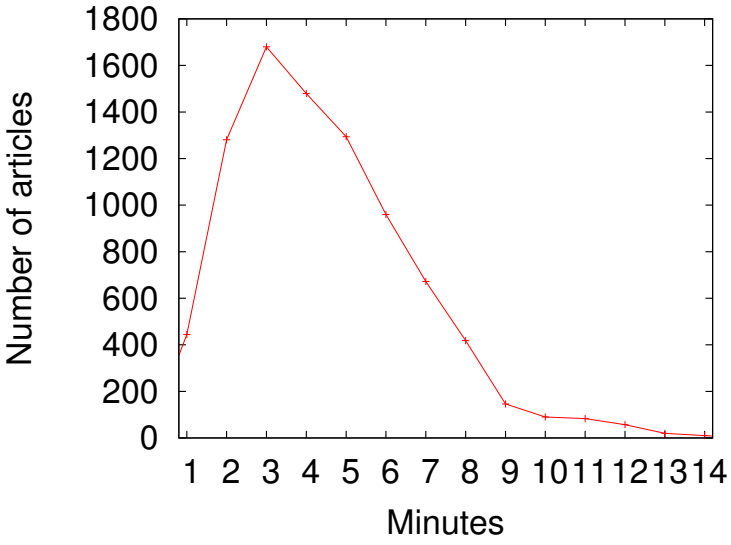
Data Size

	Rows	Data	Data + Index
cluster_locs	4.0M	8GB	15GB
clusters	2.8M	1.9GB	4.5GB
doc_locs	25.8M	5GB	12GB
docs	6.3M	20GB	81GB
doc_text	5.5M	83GB	87GB
entities	207M	25GB	38GB
feeds	10K	200MB	350MB
gazetteer	8.0M	1.5GB	4.2GB
media	1.4M	4GB	9GB

- `doc_text` is largest at 83GB
  - Not surprising as contains full text of articles in database
  - But `docs` rivals it when including index space as `docs` has many more columns and indexes on these columns, including a full-text index on the `snippet` attribute for keyword searches
- `doc_locs` and `entities` have many rows, reflecting the many locations and other entities found by NewsStand's geotagger and other processing modules

Backend Processing





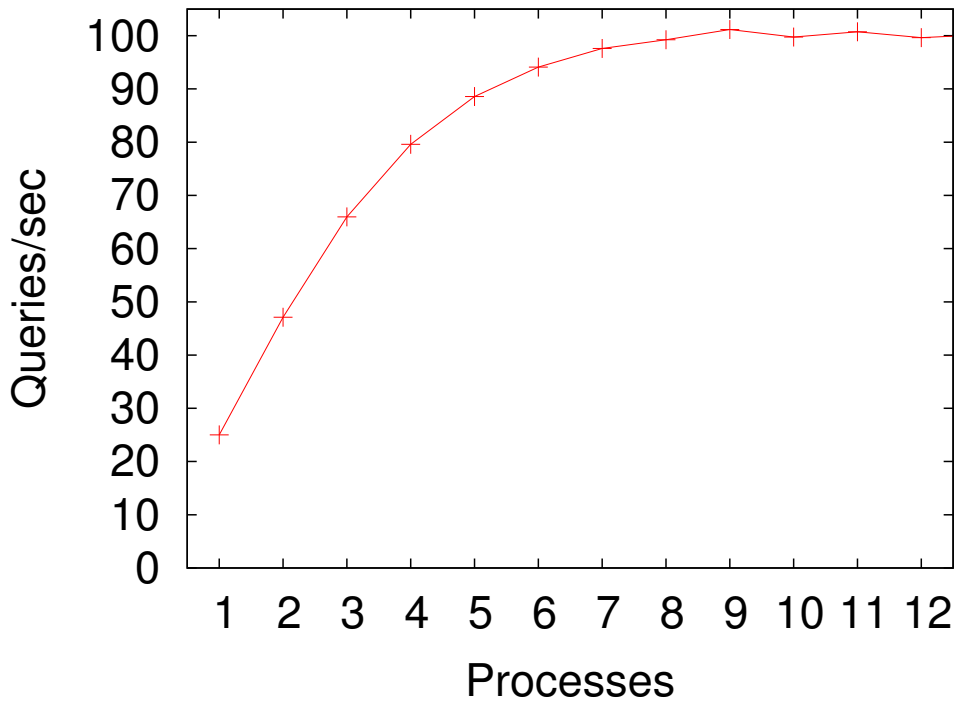
- Order of minutes backend processing to ingest batch (up to 100 but often less) of articles into database
- Most articles appear in NewsStand's web interface within 3–5 minutes

Processing time in modules per document (secs.)

Downloader	1.024	Geotagger	2.961
Cleaner	0.098	People finder	0.535
Clusterer	1.648	Disease finder	0.125
Topic classifier	0.047	Media extractor	0.166
	Total	6.604	

- Total of 6.6 seconds per document
- 85% of time in downloader, clusterer, and geotagger as make many database queries
- Can increase throughput by starting more instances of modules

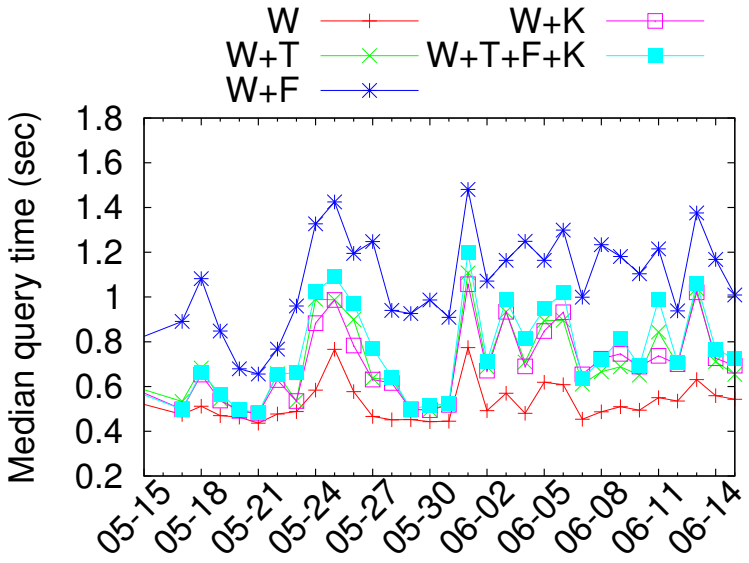
Window Query Performance



- Find k (200) clusters with highest scores in random windows
- Each query is a process and increased number until number of queries per second hit a ceiling of about 100
- Assuming user can generate 2 queries per second (via continuous actions of scrolling, panning, and zooming) implies about 50 users at a time
- k=200 which is used by NewsStand and reasonable given the limited bandwidth and screen space

Window and Constraints Query Performance

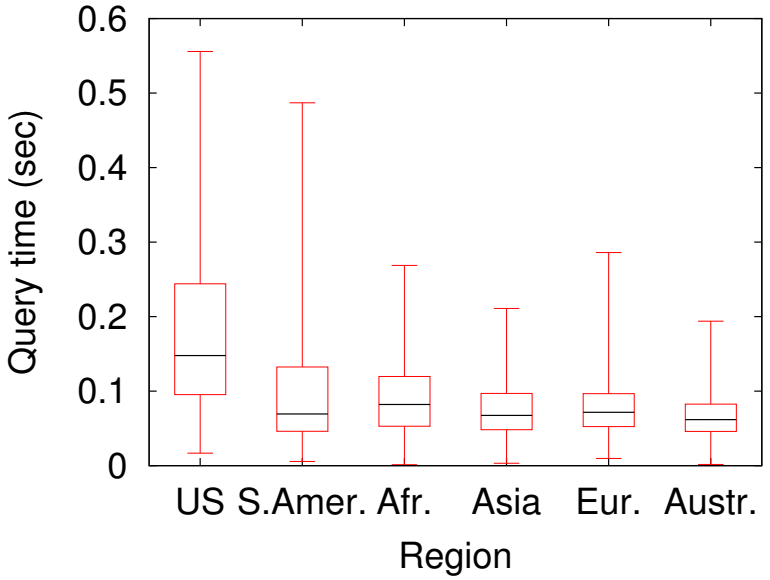
- Add constraints to top k random window (W) query:



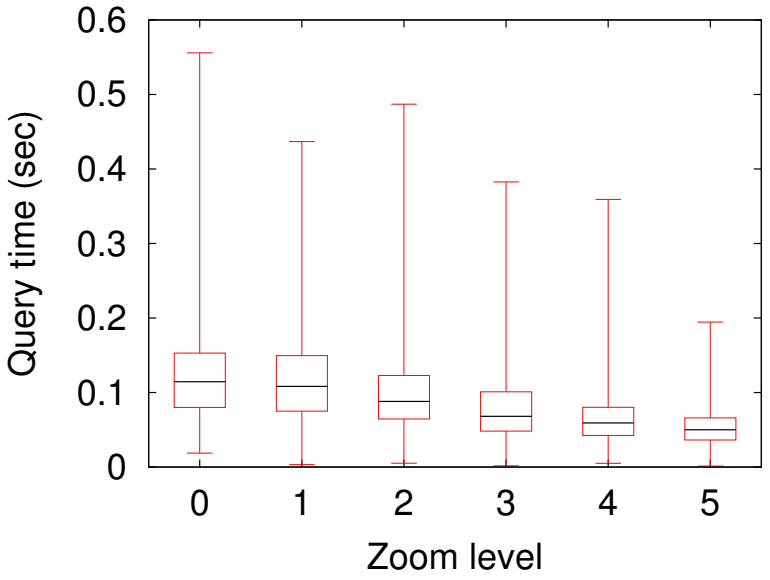
- Topic (T): Retrieve clusters relevant to a given topic (e.g., “Business”, “Sports”)
- Feeds (F): Retrieve clusters with articles from a set of news feeds
- Keyword (K): Retrieve clusters with articles relevant to a keyword
  - Only index keywords in headlines
  - Too large if index on all terms
  - Future: Index more or on first part of article text (pyramid)

- Generate queries by looking at NewsStand’s query log files
- Repeat every 5 minutes for a month
- Record daily median query time (secs.)
- Results
  - Pure W was fastest meaning that query slowed by added constraints
  - Adding feeds constraint was slowest as more data to retrieve
  - Query performance relatively consistent across dates

Geographic Region and Zoom Performance



- Random window queries over geographic regions
  - More time in US regions as more sources



- Random window queries over different zoom levels
  - Faster as zoom in more as less clusters

Concluding Remarks and Future Work Directions

- Cloud computing implementation
- Multiple cache databases
- More extensive keyword indexes
- Improved geotagging toponym recognition based on previous errors
- Informed caching strategies
  - Based on IP address of user
  - Based on prior window queries