Summary of Goods

Organizing Google’s Datasets

Abstract

- Goods is a project to rethink how we organize structured datasets at scale in an environment where team uses diverse and their specific way to produce dataset and where no centralized system for storing and querying them.

- What Goods does :
  - it collects metadata (e.g. timestamp , owners etc.)
  - find relationships among datasets.
  - exposes it to engineers to find datasets within company, to monitor datasets, to annotate datasets and to analyze relationship between them

Introduction

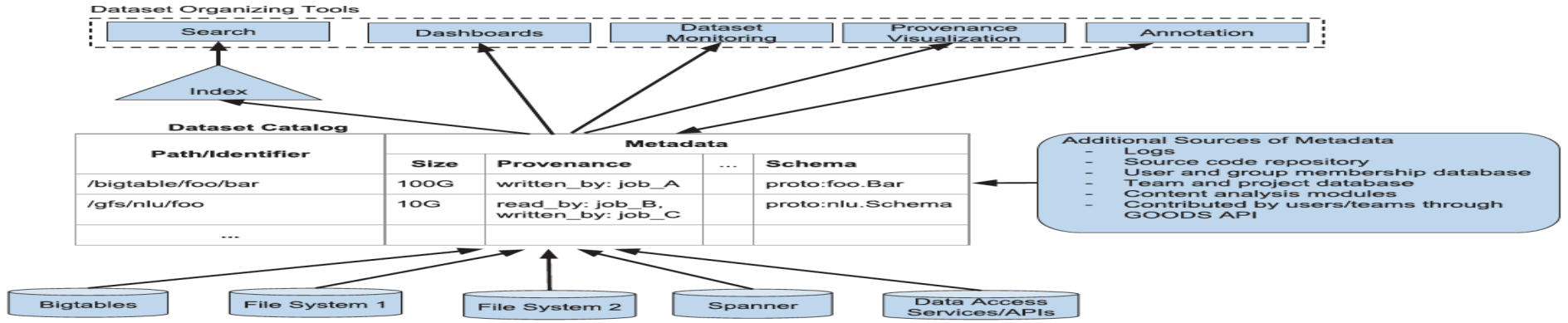

Figure 1: Overview of Google Dataset Search (Goods). The figure shows the Goods dataset catalog that collects metadata about datasets from various storage systems as well as other sources. We also infer metadata by processing additional sources such as logs and information about dataset owners and their projects, by analyzing content of the datasets, and by collecting input from the Goodsusers. We use the information in the catalog to build tools for search, monitoring, and visualizing flow of data.

- Large enterprises allow Data scientists and engineers to consume and generate data in an unfettered manner. As a result lots of data are generated. These data are assets and their management is imperative.
- Two ways to manage data:
  - EDM (enterprise data management): necessitates stakeholders to embrace its approach to publish, retrieve and integrate their datasets.
  - Alternative is to enable complete freedom in generating and consuming datasets and find right data in post-hoc manner. Goods is such a post-hoc system.
  - Goods provide Dashboard, Datasets feature monitoring, Dataset provenance, search , profile page and facilitates annotating of datasets.

Challenges

- Scale :
  - Goods catalog indexes around 26 billion datasets excluding restricted , uninteresting datasets and normalizes path to reduce redundancy.
  - “n-square” problem
- Variety:
  - Datasets are stored in many formats (txt, csv etc.) and storage systems (GFS, DB servers etc.).
  - Prioritize datasets as cost of metadata extraction varies a lot with type and size of dataset
- Churn of the catalog entries:
  - 5% of datasets are deleted and added everyday.
  - Prioritize datasets. Ignore transient datasets unless it serves as a link between non-transient datasets.
- Uncertainty of metadata discovery:
  - Dataset itself does not reference the specific protocol buffer that describes its content.
  - Goods tries to uncover this implicit association through several signals which are inherently ambiguous.

- Computing dataset importance:
  - what makes a dataset important is hard to answer.
  - Ranking in Goods is significantly different from the Web search setting (e.g. no anchor text here)
- Recovering DS semantics:
  - Semantics information is extremely important for searching, ranking and describing DS.
  - However, identifying the semantics from raw data is hard as there is rarely enough information in data to make this inference.

The Goods Catalog

The catalog contains entry for each dataset that Goods discovers by crawling different storage system.

Metadata: Types of Metadata in Goods Catalog:

- Basic metadata: It includes timestamp, file format, owners, size etc.
- Provenance:
  - Goods identify and populate the provenance info through analysis of production logs (which job reads each datasets). It then creates the transitive closure of the graph connecting datasets and jobs.
  - As number of data access events in logs can be extremely high, there is a trade off between completeness of provenance associations and efficiency of processing.
- Schema:
  - Schema is another core type of metadata that helps us understand a dataset.
  - Nearly all records within structured datasets at Google are encoded as serialized protocol buffers.

- Content summary:
  - Goods finds most frequent tokens or tokens which can summarize the content.
  - Goods also collects fingerprint which can be used to find if contents of two datasets are similar. Two important algorithm used are: HyperLogLog algorithm and Locality sensitive hashing.
- User provided annotations :
  - These descriptions are helpful in ranking and filtering of datasets.
- Semantics:
  - Goods combine several noisy signals to derive metadata about semantics.
  - For example: Goods can examine the source code that defines the protocol buffer and extract any of the attached comments.
- Other metadata that Goods collect are id of teams that own dataset, description of project to which dataset belongs and history of changes of metadata of the dataset.
- Finally, Goods allows other teams to add their own types of metadata.

Organizing datasets into clusters:

- We often see different versions of datasets are generated regularly, DS are replicated across data centers, DS are sharded into smaller datasets for faster loading and so on.
- Identifying natural clustering helps us to
  - provide logical-level abstraction
  - save cost of metadata extraction
- The paths of the datasets often give hints on how to cluster them. Let /dataset/2015-10-10/daily_scan be the path for one of the instances of DS.
  - /dataset /2015-10-<day>/daily_scan represents all instances from October 2015.
  - /dataset/2015-<month>-<day>/daily_scan represents all instances from year 2015.
- By composing hierarchies along different dimensions we can construct granularity semi-lattice and each node can be thought of as a clustering choice.

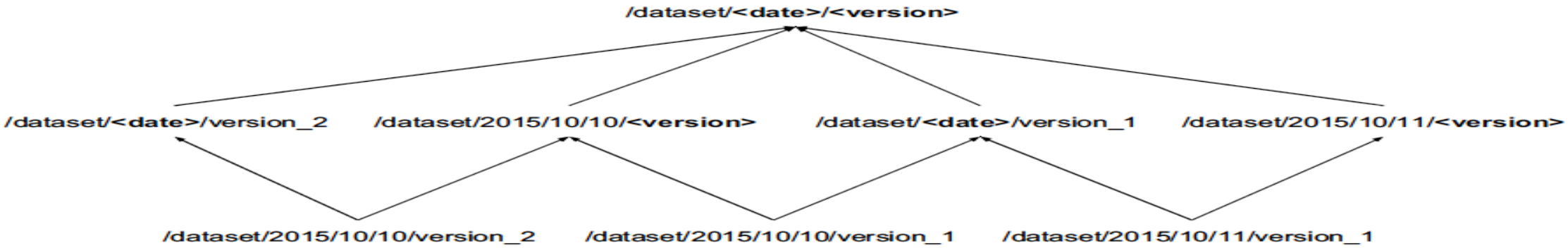

Figure 2: An example of abstraction semi-lattice

- Above figure shows an example of such semi-lattice obtained by composing two hierarchies , along date and version number.
- Goods cluster on top-most element to keep number of cluster entries low.
- Metadata of a cluster is obtained by aggregating the metadata of individual member.

Backend Implementation

- Catalog storage
  - Backed by BigTable
  - Data accessed by batch jobs are kept in separate column family
  - 2 kinds of metadata: 1)Metadata for DS 2) status metadata
- Batch job performance and scheduling
  - Jobs run independently and are optimized to finish within 24 hours
  - Sometimes interdependency among modules e.g. computing fingerprint and schema analyzer
  - When large influx of DS then priorities DS and create two job instances : one for high priority DS (finishes quickly) and other job targets all DS.
- Fault tolerance
  - Triggers retries for modules that ended with an error.
  - Sandboxes dangerous jobs in separate process and uses a watchdog.
  - Goods replicates catalog at multiple geographical locations.

- Garbage collection of metadata
  - Three constraints
    - Remove rows only when “the dataset has been deleted from the storage system, and its most recently updated provenance information has been processed by a transitive provenance linker module that finished successfully.”
    - Must not create dangling rows.
    - All other modules run independently of and simultaneously with garbage collector.
  - Garbage collection occurs in 2 phases
    - Using first constraint (mentioned above) garbage collector puts tombstone on a row.
    - After 24 hours garbage collector inspects the row again for tombstone. If tombstone is still present then remove that row.

Front end

- Dataset profile page
  - Goods exports metadata to profile page
  - Profile page is one-stop shop for inspection of datasets
  - Cross links, between metadata and jobs that generated it, are present
  - Profile page also provides access snippets to access the contents of datasets.
- Dataset search
  - Backed by conventional inverted index for document retrieval.
  - Indexing tokens are derived from a subset of dataset's metadata. For example, indexing tokens for path are generated by breaking up path along common separators and associating each resulting token with its position in path.
  - Heuristics for scoring function:
    - Importance of DS depends on its type.
    - Importance of keyword match depends on index section.
    - Lineage fan-out: favors DS with many reading jobs.
    - A dataset that carries owner-sourced description is likely to be important.
- Team Dashboards
  - one-stop shop for displaying all the DS generated by team along with metadata e.g. health metric.
  - Provides means to monitor datasets and fire alerts.

Lessons learned

- Evolve as you go
  - Several new use cases arose e.g. re-find dataset, audit protocol buffer, understand legacy code etc.
- Use domain-specific signals for ranking
  - Provenance relationship provide a strong domain specific ranking signal.
- Expect and handle unusual datasets
- Export data as required
  - Export the catalog data to a suitably specialized engine , if required.
- Ensure recoverability
  - BigTable retains snapshots over several days.
  - Dedicated process to snapshot high-value dataset in a separate catalog.
  - Another process to replicate metadata that powers profile page.
  - Use Goods dataset monitoring service for catalog itself.

Related work

- Data-lake management of IBM is similar to Goods
- Goods can be thought as concrete implementation of Dataspace
- Datahub is a version-management system for datasets. Similar in spirit to SVN for codes.
- Spyglass and Propeller are search systems but unlike Goods they are on single storage system.
- Examples of provenance management tools are PASS and Trio. But unlike Goods they assume provenance information is given or can be retrieved by some other tools.

Conclusion and future work

- Based on user's feedback it appears that lots of work need to be done in search
- Integrating Goods with google knowledge graph will provide more information on missing metadata about a dataset.
- Exploring a combination of post-hoc and non post-hoc manner.