Finding a Needle in a Field of Haystacks

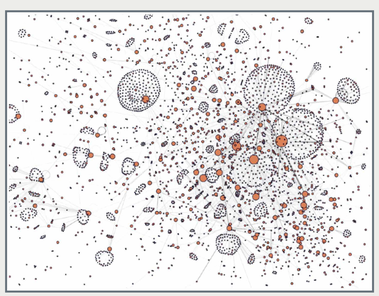
Lightweight Metadata Search for Large-Scale Distributed Research Repositories

Anna Blue Keleher — University of Maryland
Kyle Chard (Advisor) — University of Chicago
Ioan Raicu (Advisor) — Illinois Institute of Technology
Ian Foster (Advisor) — University of Chicago
Alex Orhean (Advisor) — Illinois Institute of Technology

1 ABSTRACT

- Fast, scalable, and distributed search services are increasingly available to organizations with large volumes of data to index, but often require that data be located centrally [1]
- Globus provides access to a distributed network of storage systems for data access and transfer; however there is no way to search across its collection of over 10,000 active endpoints
- Our solution: a two-level tree of lightweight metadata indexes that retains summary terms able to reference data while drastically reducing the amount of data stored centrally by upwards of 96% — a "second-level index" (SLI)

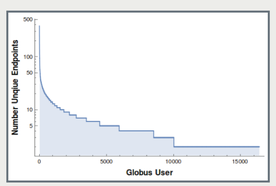
2 GLOBUS



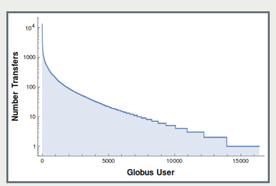
Globus file transfer network. Nodes represent endpoints, scaled by data size, edges represent transfers (length is inversely proportional to transfer volume).

Globus by the Numbers

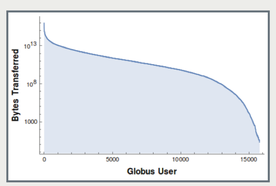
- 300 PB transferred
- 50 billion files processed
- 1 PB — largest single transfer to date
- 10,000 active endpoints
- 78,000 registered users
- 10,000 active users/year



Unique endpoints per Globus user



Number of transfers per Globus user



Bytes transferred per Globus user

Why metadata?

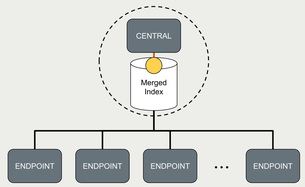
- Globus connects storage systems, many of which contain files that are not full text or easily indexable
- In such storage systems, files are often characterized by long names and directory paths rather than contents
- Metadata is much easier to work with at scale — and Globus is huge

Which data?

- 37 largest public Globus endpoints (2.7 TB total)
- NERSC filesystem dump, similar to large private endpoints (164.6 TB total)

(Globus network images courtesy of William Agnew, Georgia Institute of Technology.)

3 DESIGN/SETUP

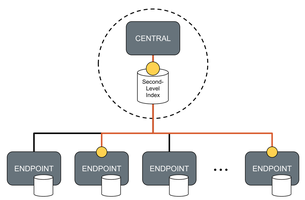


Traditional 1: single central index

DESCRIPTION
- Central index is result of merging indexes from each endpoint

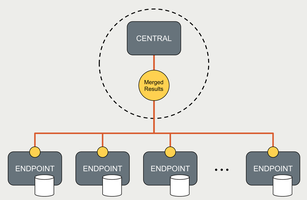
ADVANTAGES
+ No endpoints must be queried individually
+ No need to centrally merge results

PROBLEMS
- Heavy storage burden on central index: contains all references to docs (grows too fast, difficult to synchronize)



Our Method: hierarchical hybrid featuring "collapsed" second-level index (SLI)

- SLI references endpoints, not docs, and contains a summary subset of terms
+ Some storage burden on endpoints, but still very low per endpoint
+ Lower storage burden on central servers
+ SLI returns a smaller subset of endpoints to which queries must be distributed



Traditional 2: fully distributed index

DESCRIPTION
- Indexes kept on each endpoint
- Central servers distribute query to each endpoint, then retrieve/merge results

ADVANTAGES
+ More space total used by indexes than in Trad. 1, but burden split among endpoints

PROBLEMS
- Most search terms are likely to correspond only to a small subset of endpoints

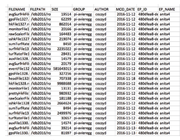
Query path · Globus servers · Search results · Index

Indexing with Xapian

- Xapian [3] is an open-source indexing engine for full-text search with strong Python support
- We add new tokenization capabilities to allow it to interpret terms from filepaths, camel-case filenames, etc. `"IAmManyTerms.csv" -> F:'I','Am','Many','Terms'; E: 'csv'`
- Terms in the SLI are aggressively stemmed and numerical fields (e.g., size, modification time) are collapsed into ranges

NERSC FS Data Prep

- File system dump from February 2017
- We crawl FS to extract metadata
- Divide metadata by "project," then bin into 8 equal-sized shards
- Distribute across 8 VM "endpoints"
- Binning by project prevents "endpoints" from being too similar



Sample metadata

4 RESULTS (PUBLIC ENDPOINTS)

SLI Performance

Model	Size of Index at Location (MB)		
	Central	Endpoints	Per-EP Avg.
Centralized	10,369.71	0	0
SLI	**239.80**	13,001.15	351.38
Distributed	0	13,001.15	351.38
	Size reduction of central index		**97.69%**

SLI Performance (above)

- The SLI model uses more space total, but drastically reduces storage overhead on central index

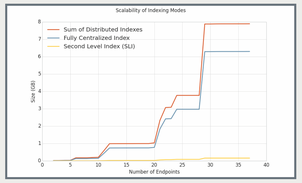
SLI Scalability (right)

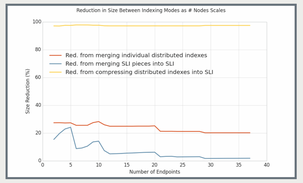
- x-axis: number of endpoints (in arbitrary order) used to build the merged index
- "Reduction" graphs demonstrate that the shift to the SLI is consistently beneficial

SLI Scalability

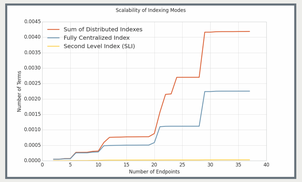
STORAGE GROWTH — REDUCTION

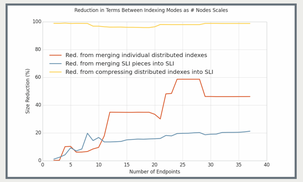
SIZE OF INDEX





OF TERMS

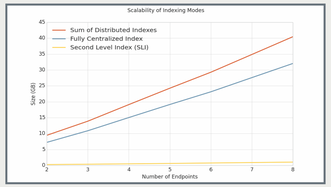




5 RESULTS (NERSC FS)

STORAGE GROWTH

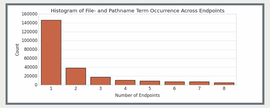
SIZE OF INDEX



SLI Performance

Model	Size of Index at Location (MB)		
	Central	Endpoints	Per-EP Avg.
Centralized	30,552.83	0	0
SLI	**969.83**	38,597.14	4,822.39
Distributed	0	38,597.14	4,822.39
	Size reduction of central index		**96.83%**

Term Dist. Over Endpoints



- The majority of terms indexed from filename and pathname metadata appear only on a sub-set of the endpoints
  - 60.3% of terms exist on one
  - 76.3% of terms exist on ≤ two
  - Only 2.1% of terms exist on all
- NERSC distribution stats are conservative estimates due to similarity of NERSC "endpoints"
- In general, few terms will exist on all endpoints, making the SLI's ability to target which indexes to query extremely valuable

6 CONCLUSIONS

- Our second-level index (SLI) finds a workable compromise between the two traditional large-scale indexing architectures
- The SLI model dramatically reduces the size of a central merged index and reduces query overhead by selecting only relevant endpoint-level indexes
  - By using two levels and referencing endpoint-level indexes, the central index can avoid storing all terms, instead only extremely stemmed terms and numerical ranges
  - Because full indexes are retained on endpoints, search is still accurate once redirected to those endpoints
  - Recall stays ~1 because queries at SLI level are stemmed the same as terms. Precision is higher than Traditional Method 2.
- The SLI scales more efficiently than a traditional index

7 ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation grant NSF-1461260 (BigDataX REU).

8 REFERENCES

[1] Reaz Ahmed and Raouf Boutaba. 2011. A survey of distributed search techniques in large scale distributed systems. IEEE Communications Surveys & Tutorials 13, 2 (2011), 150–167.
[2] I. Foster. 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. IEEE 15, 3 (may 2011), 70–73.
[3] Vesna Hassler. 2005. Open source libraries for information retrieval. IEEE Software 22, 5 (2005), 78–82.