Finding a Needle in a Field of Haystacks: Metadata Search for Distributed Research Repositories

Anna Blue Keleher
University of Maryland
bluek@cs.umd.edu

Kyle Chard, Ian Foster
(Advisors)
University of Chicago
chard@uchicago.edu
foster@cs.uchicago.edu

Ioan Raicu, Alex Orhean
(Advisors)
Illinois Institute of Technology
iraicu@cs.iit.edu
aorhean@hawk.iit.edu

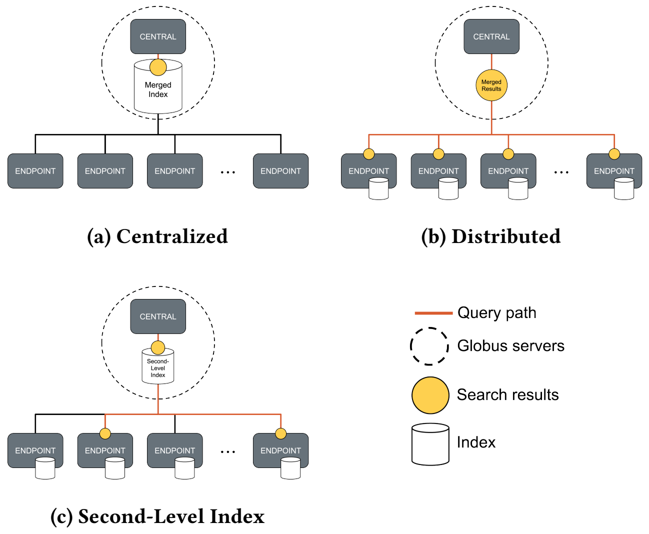
ACM Reference format:
Anna Blue Keleher, Kyle Chard, Ian Foster (Advisors), and Ioan Raicu, Alex Orhean (Advisors). 2017. Finding a Needle in a Field of Haystacks: Metadata Search for Distributed Research Repositories. In *Proceedings of SC '17, Denver, CO, USA, November 12-17, 2017,* 2 pages.
https://doi.org/

1 INTRODUCTION

Free-text search across a billion-file filesystem is an old and often-addressed problem [5, 8]. However, when that system is distributed and multiplied by several thousand, traditional methods become infeasible. This is the landscape presented to users of Globus [3], a service that facilitates high performance and reliable data access, transfer, and synchronization across a network of more than 10,000 research filesystems and repositories (called "endpoints"). The semi-structured nature and scale of these endpoints makes manual search difficult for users, especially as they gain access to more endpoints over time.

To avoid having to build a new access-specific inverted index for each query, we consider indexes built across the entire Globus network, irrespective of access privileges (which may be subsequently accounted for when filtering results). We also limit the scope of search to file metadata (i.e., a "find" query) for two reasons: first, it makes indexing the entire network (centrally or otherwise) feasible by reducing the data sizes stored; and second, it is appropriate for many research repository file types which contain little free text [7].

Traditional methods suggest two approaches to search in such a model: a single complete centrally-stored index (the *Centralized Model*) [2, 6]; and distinct indexes at each endpoint (the *Distributed Model*) [1]. Search engines like Solr [9] and ElasticSearch [4] can accomplish the former by maintaining complete indexes at a centralized location; however, scaling to the Globus network requires significant investment in infrastructure. The alternative is building indexes at each endpoint and implementing support for distributed querying. While this approach reduces central storage requirements, it adds significant query overhead as many indexes must be queried.



(a) Centralized **(b) Distributed**

(c) Second-Level Index

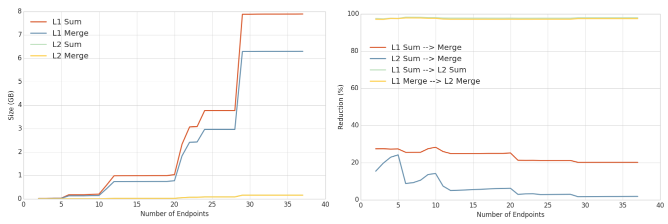
Figure 1: The three indexing architectures

Our solution provides a compromise by introducing two levels of indexes composed of many endpoint-specific indexes and a single, approximate, central *Second-Level Index* (SLI). The SLI is much smaller than the full centralized index because it reduces the number of terms in the index (through aggressive stemming and range consolidation) and the number of objects referenced for each term (endpoints rather than documents). Queries are then only redirected to the relevant subset of endpoints. The three index models are illustrated in Figure1.

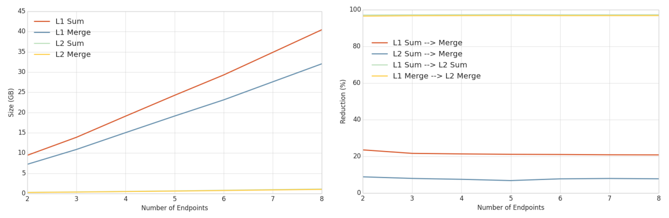
2 METHODOLOGY

Irrespective of which indexing model is used, each endpoint must be crawled to compile a comprehensive set of metadata. We use the Xapian search engine [6] for indexing and extend its tokenizer to parse terms from metadata-specific string formats and numerical fields.

In the SLI model, we construct a standard endpoint-level index and then collapse fields and records into the SLI. For example, string fields (e.g., name, pathname) are split by separators (e.g., underscore, camel-case) and then aggregated into common tokens. Numerical fields (e.g., size, modification date) are collapsed into ranges. Queries at the SLI are stemmed in the same way as its index terms so that they are fully represented by the index, eliminating false negatives. Full queries are then passed to identified endpoint indexes.



(a) Scalability of Public Endpoints



(b) Scalability of NERSC Filesystem

Figure 2: Scalability of Size of Indexing Modes and Reduction Between Modes

3 RESULTS

We explore the performance of the three indexing models using two different sets of real Globus data: the 37 largest public Globus endpoints (as of June 2017), which are generally small and diverse; and a snapshot of a NERSC filesystem, representative of large, private Globus endpoints. For the NERSC dataset, we simulate a Globus network structure by grouping documents by path proximity, and binning them into eight roughly equal shards on separate virtual machine instances that communicate with a central node.

Table 1 shows the index size for each model and dataset. For both datasets, the central SLI index is more than 96% smaller than that required in the Centralized Model. Because the SLI model also stores traditional indexes on each endpoint, its aggregate space requirements are slightly higher, but the burden on any single node is greatly reduced.

Table 1: Index Size Comparison Between Models

Public Endpoints

	Size of Index at Location (MB)		
Model	Central	Endpoints	Per-Endpoint Avg.
Centralized	10,370	0	0
SLI	**240**	13,001	351
Distributed	0	13,001	351
Size Reduction of Central Index			**97.69%**

NERSC Filesystem

	Size of Index at Location (MB)		
Model	Central	Endpoints	Per-Endpoint Avg.
Centralized	30,553	0	0
SLI	**970**	38,579	4,822
Distributed	0	38,579	4,822
Size Reduction of Central Index			**96.83%**

In order to evaluate the SLI model's scalability, we track the sizes of the centralized index and SLI as endpoints are added. As shown in Figure 2, both the aggregate (sum) and merged indexes for the SLI model grow at a much lower rate than the traditional centralized index, and the reduction percentage for the SLI model does not decay as data grows.

We also note that the incidence of terms existing on all endpoints is low, even between the artificially similar NERSC"endpoints." We conclude that the SLI model improves on the Distributed Model whenever the SLI accurately narrows the set of relevant endpoints, which Figure 3 indicates is likely to be common.

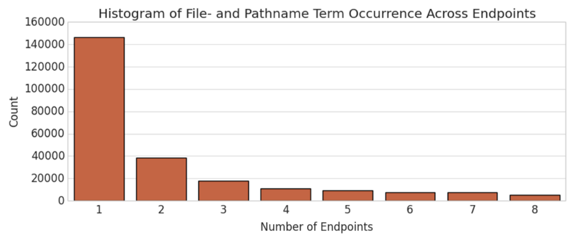


Figure 3: Frequency of filename or pathname terms occurring on multiple endpoints

4 CONCLUSION

We present a second-level index (SLI) model as a compromise between the two traditional large-scale indexing architectures. We evaluate the effectiveness of the approach on data across a collection of Globus endpoints. We find that the SLI algorithm stores far less data at the central node than a purely centralized approach while also greatly reducing search overhead relative to a purely distributed approach. Thus this approach makes search over the complete Globus ecosystem achievable and scalable.

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] Miguel Costa, Daniel Gomes, Francisco Couto, and Mário Silva. 2013. A survey of web archive search architectures. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 1045–1050.
[3] I. Foster. 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *IEEE* 15, 3 (2011), 70–73.
[4] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide.* OReilly.
[5] Ed Greengrass. 2000. *Information retrieval: A survey.* University of Maryland.
[6] Vesna Hassler. 2005. Open source libraries for information retrieval. *IEEE Software* 22, 5 (2005), 78–82.
[7] Andrew W Leung, Minglong Shao, Timothy Bisson, Shankar Pasupathy, and Ethan L Miller. 2009. Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems.. In *FAST*, Vol. 9. 153–166.
[8] A. Narang, V. Agarwal, M. Kedia, and V. Garg. 2008. IBM Research Report: Scalable Algorithm for Distributed In-Memory Text Indexing. (2008).
[9] Dikshant Shahi. 2015. Apache Solr: An Introduction. In *Apache Solr*. Springer, 1–9.