Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 43–52

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

OpenWebSearch.eu - Building an Open Web Index on EuroHPC JU Infrastructures

Michael Granitzer[a,*], Mohamad Hayek[b], Sebastian Heineking[c], Gijs Hendriksen[d], Martin Golasowski[e], Michael Dinzinger[a], Saber Zerhoudi[a]

[a] *Chair of Data Science, University of Passau, Passau, Germany*
[b] *Leibniz-Rechenzentrum (LRZ), Munich, Germany*
[c] *University of Leipzig, Leipzig, Germany*
[d] *Radboud University, Nijmegen, The Netherlands*
[e] *IT4I, VSB – TU of Ostrava, Ostrava, Czech Republic*

Abstract

The OpenWebSearch.eu project aims to develop an Open Web Index (OWI), an openly accessible data structure that supports the creation of web search engines. Building such an index requires a data- and compute-intensive pipeline for cleaning, pre-processing, enriching and indexing large amounts of web data. Beyond search, the availability of clean and preprocessed web data is also crucial for fields like web analytics and generative AI. This paper presents our approach to constructing the OWI using High-Performance Computing (HPC) resources from both EuroHPC JU and non-EuroHPC JU data centers. We contribute in two main areas: first, by detailing the development of pre-processing and indexing pipelines embedded in HPC workflows; and second, by describing the iRODS-based federated storage infrastructure and the LEXIS[1] platform that will manage the cross-data centre workflows and facilitate the publication of the OWI as daily datasets. During the alpha phase, from October 2023 to April 2024, we processed approximately 76 TB of web data, encompassing over 2 billion URLs. By addressing the challenges of large-scale web data processing and retrieval, this work lays the foundation for an innovative, competitive, and transparent web search ecosystem, while also supporting the development of European generative AI solutions.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Web Data and Web Search; Open Web Index; Information storage and retrieval

1. Introduction

The Web is a vital resource for various applications, including search engines, data analytics, and generative AI. However, harnessing web data presents significant challenges due to high demands on hardware, technical complexity,

[1] https://portal.lexis.tech

* Michael Granitzer Tel.: +49-851-509-3300.
E-mail address: michael.granitzer@uni-passau.de

1877-0509
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.259

legal constraints, and data quality issues. These barriers particularly affect small innovators and researchers, making it difficult for them to compete with major industry players. The resulting lack of competition reinforces the dominance of a few large search engines, stifling innovation and reducing diversity in search services, analytics, and generative AI.

To address these challenges, we present the Open Web Index (OWI) [4, 6], an openly available index of the web created through a collaboration of High Performance Computing (HPC) centres in the OpenWebSearch.eu project. Creating an index of the Web usually requires large-scale crawling of web content, data cleaning, data preprocessing, deduplication, and indexing. The sheer scale of the data presents the biggest challenge when indexing the web, particularly in terms of computing and storage resources. Just for comparison: it is assumed that Google's index contained roughly 400 billion web pages in 2020, according to information from a lawsuit[2], which would equate to several 100 PB in storage size. Clearly small single organisations, research institutes, and even larger companies cannot provide the necessary resources for creating such an index.

By utilising Europe's HPC infrastructure, particularly resources provided by EuroHPC JU, the OpenWebSearch.eu project aims to create an openly available index of the Web, called the Open Web Index. While the index size will remain smaller than commercial indices, it is the first openly available index following open source and open data principles. Although the EuroHPC JU initiative provides significant HPC infrastructure for compute-intensive tasks, using web data for search, analytics, or AI also presents data-centric and IO-centric challenges. These include indexing web data for search, preprocessing data (including natural language processing for semantic enrichment), and computing AI-based embeddings for dense retrieval and Retrieval Augmented Generation (RAG).

This paper presents our current pipelines and achieved results for creating an Open Web Index on a collaborative network of HPC centres, both within EuroHPC JU - particularly IT4Innovations National Supercomputing Centre (IT4I) and CSC - IT Center for Science (CSC) - and outside of EuroHPC JU - particularly the Leibniz Supercomputing Center in Munich (LRZ), CERN and the German Aerospace Center (DLR). Specifically, we address the following contributions:

- Development of robust preprocessing and indexing pipelines using HPC resources for converting crawled data into a shareable and extensible index.
- Cross-data center execution of HPC workflows and dataset management based on the LEXIS Platform[3] and central authentication and access management via B2ACCESS.
- A federated, iRODS[3]-based storage system for storing and sharing workflow outputs across HPC centers.
- Tools for pulling, pushing and querying the index datasets computed via HPC resources, promoting collaborative management of web data and the Open Web Index.

These contributions collectively address the challenges of processing large-scale web data and set the foundation for an open, collaborative web search infrastructure. However, our work also goes beyond creating an open web index through the establishment of federated data storage across HPC data centres, a single point of execution for HPC jobs across data centres, and data set publishing and management tools for managing very large data sets.

The paper is structured as follows: In section 2 we start by describing the vision of an Open Web Index and describe the application scenario. Afterwards, we give an overview on the HPC workflows 3 including individual components, steps and the underlying storage concept. Section 4 outline the achieved results so far in terms of HPC-utilisation and data set size, while section 5 concludes the work.

2. Application Scenario: Vision of an AI-powered Open Web Index

We envision the Open Web Index (OWI) as a distributed information system built on a federated storage infrastructure. This system allows search engines and web data-centric applications to retrieve data from storage systems

[2] https://zyppy.com/seo/google-index-size/
[3] Integrated Rule-Oriented Data System https://irods.org/

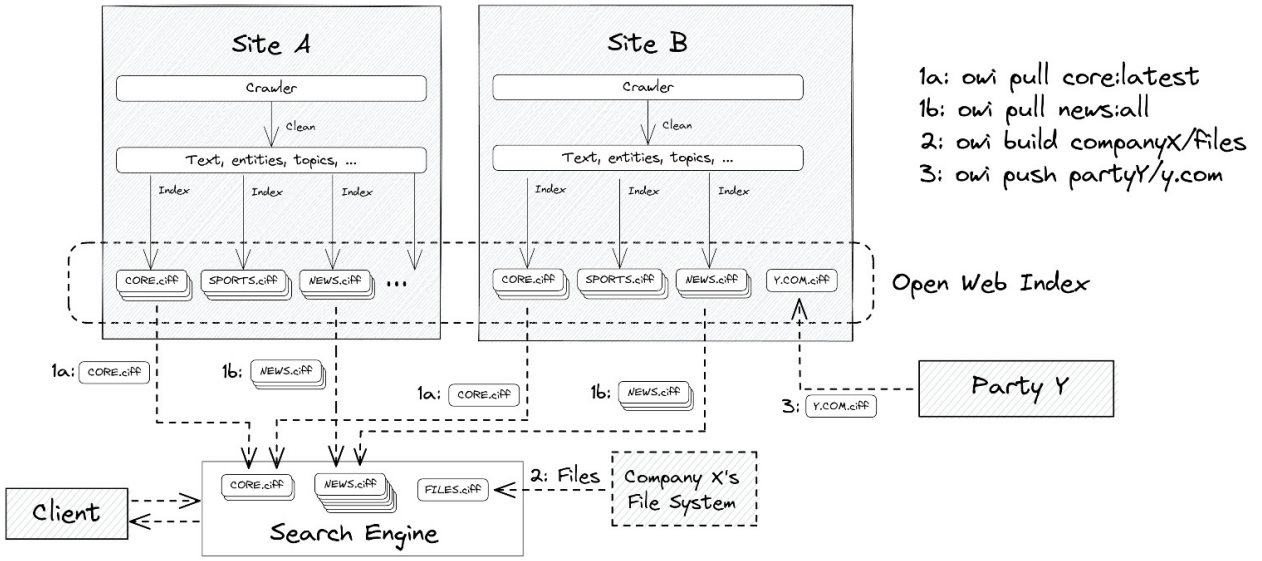

Fig. 1. General architecture of the OWI and its interaction with search engines for index retrieval.

through flexible selection methods. Users can choose horizontal slices based on date and language or vertical slices by selecting specific attributes of interest (e.g., title, plain text, and URL of a web page). This "slice & dice" concept enables search engines to obtain data at regular intervals according to their specific requirements.

Pre-computed indices are provided in the Common Index File Format (CIFF) [9], ensuring compatibility with existing open-source search engines such as Lucene[4], (Py)Terrier [10, 12], and PISA [11]. Beyond traditional inverted indices stored in CIFF format, modern AI-based retrieval applications typically rely on dense retrieval of sub-document units for use with Large Language Models, a technique known as Retrieval Augmented Generation (RAG) [8]. Dense retrieval, particularly in the RAG context, necessitates the computation of dense vector embeddings, which usually involves applying (Large) Language Models to chunked web text.

We further envision that specifications for search engines can be stored in a descriptive manner, detailing not only the required slices and index requirements but also search and configuration modalities (e.g., ranking mechanisms, search and database backends). Storing search engine declarations in this way could conceptually yield a system similar to Docker Hub, but specifically for web search and web analytics applications.

Beyond data retrieval, the federated storage would also allow users to push data, such as embeddings for dense retrieval or annotations of web content. This push-pull-slice paradigm for web data would form the basis for collaborative management of web data on top of HPC-backed federated data storage. Figure 1 illustrates the general architecture of the OWI and its interaction with search engines.

3. High-Performance-Computing Pipelines for creating an Open Web Index

High-Performance Computing (HPC) centers play a crucial role in creating an Open Web Index by providing the necessary computational power for big data processing. Web data usually consists of terabytes to petabytes, necessitating meticulous planning of storage strategies. Furthermore, the compute and storage resource demands for building an OWI can exceed the capacity of a single HPC center, especially when considering potential future extensions to generative AI. In order to pool the necessary resource for building the OWI, we therefore aim to coordinate workflows across multiple data centers and have a federated storage across HPC centers.

Figure 2 provides an overview of the workflows in one data center, consisting of the following stages:

[4] https://lucene.apache.org/

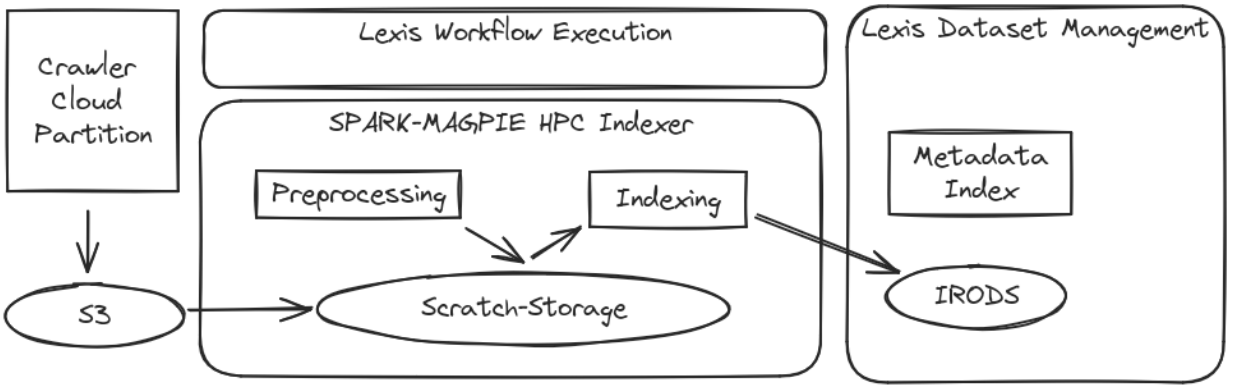

Fig. 2. HPC Workflows for a single data center

1. **Data Ingestion**: Data is ingested via a separate crawling system [2] delivering approximately 2 TiB per day. Data is staged in S3, with crawlers currently running at all participating data centers, coordinated via a central frontier at CERN. This approach allows for the distribution of crawling and HPC load among different HPC centers.
2. **Workflow Execution**: Workflows are managed and executed using the LEXIS Platform and run across three HPC centers: IT4I, LRZ, and CSC. Workflows begin by moving S3 data to cluster scratch space. The processing is run on an Apache Spark cluster that we create on the HPC infrastructure using the script collection Magpie[5].
3. **Workflow Components**: The workflows consist of two main components that (1) preprocess[6] the data to create Parquet files, and (2) index and partition[7] the Parquet files. The result is a set of CIFF index files [9, 6]. Intermediate results between indexing and preprocessing are stored in scratch space and only moved to the federated iRODS storage after indexing is completed. At this stage, metadata such as size and crawl information is added to the files.
4. **Federated iRODS Storage**: iRODS serves as the backend for storing and distributing datasets. Per day and data center, we store a set of Parquet files containing preprocessing results and CIFF index files, partitioned according to year, month, day, and language of the data. The data can be consumed via the LEXIS Platform.
5. **Authentication and Access Management**: LEXIS integrates authentication via B2ACCESS, ensuring proper control over access to workflow execution and data.
6. **Tooling for Data Access**: On top of the LEXIS platform, we developed the OWI Python client *owilix*[8] which provides OWI-specific dataset management, including means for pushing and pulling datasets as well as the ability to conduct SQL queries remotely over datasets. The client uses the Python interface to the LEXIS Platform - py4lexis[9] which wraps the LEXIS Platform API to a convenient Python library.

In the following subsections, we provide more details on the individual steps.

3.1. Preprocessing

Preprocessing is the first step in the Open Web Index pipeline. It is primarily handled by two software components, Resiliparse and Resilipipe, that work in tandem to efficiently process and analyze web archive data at scale.

[5] https://github.com/LLNL/magpie
[6] https://opencode.it4i.eu/openwebsearcheu-public/preprocessing-pipeline
[7] https://opencode.it4i.eu/openwebsearcheu-public/spark-indexer
[8] https://opencode.it4i.eu/openwebsearcheu-public/owi-cli
[9] https://opencode.it4i.eu/lexis-platform/clients/py4lexis

3.1.1. Resiliparse: Core Parsing Library

Resiliparse[10] is an open-source web archive processing library and as such the foundational component for reading and parsing the web crawls. The native Python module is designed to be both efficient and robust to be able to process large amounts of web documents while handling the diversity that comes with that data. This library facilitates the rapid and safe processing of potentially malformed or malicious web content, emphasizing minimal assumptions about data well-formedness. The two main modules of Resiliparse are:

- **FastWARC**: The FastWARC library is a faster and more efficient alternative to existing WARC parsing libraries. It supports uncompressed WARCs as well as gzip- und lz4-compressed archives.
- **Resiliparse Core**: The core library offers a collection of tools to process web data. These include (1) efficient HTML parsing and DOM processing, (2) reliable character encoding detection and conversion to Unicode, (3) fast detection of common MIME types, (4) fast heuristic-based main content extraction [1], and (5) basic but fast language detection.

Resiliparse is written primarily in native C and C++ using Cython, with some parts written in Python, to offer significant memory and CPU efficiency. The tools offered by Resiliparse collectively support the extraction of plaintext and main content from web pages with high reliability and speed.

3.1.2. Resilipipe: Scalable Content Analysis Framework

Building upon Resiliparse, Resilipipe is a scalable framework implemented for cluster-based web content analysis. It is built to handle large amounts of web archive data and extracting valuable metadata that enriches the Open Web Index. Core features of Resilipipe include:

- **Cluster Deployment**: Resilipipe uses Apache Spark for parallel processing of WARC files, effectively distributing the workload across cluster nodes. In combination with Magpie and our collection of Spark deployment scripts[11] it can be easily deployed on HPC clusters with common resource managers such as Slurm or Moab.
- **Content Analysis**: The framework uses Resiliparse to efficiently read and parse WARC files. It then extracts metadata from the parsing output such as MIME types, languages, and web page categories. Advanced metadata related to geolocation, microdata, and JSON Linked Data are also extracted to enhance search engine functionalities.
- **Pass-through Metadata:** Resilipipe also allows to pass-through metadata from the crawler to indexing and storage. This becomes relevant as the crawler can provide information relevant to further processing steps, such as, for example, the fetch speed of a web page or the allowed usage pattern. Particularly important is *gen-AI flag*, which indicates whether the web page is allowed to be used with generative AI or not.
- **Modular Design**: Resilipipe supports the integration of user-created content analysis modules via a standardized interface. This allows for the extension of its capabilities based on project needs and third-party contributions. With the help of TIREx[12], The Information Retrieval Experiment platform, we can evaluate content analysis modules in a scalable and reproducible way.

3.2. Indexing

The indexing phase is a crucial step in the Open Web Index pipeline, transforming the preprocessed content into a usable, efficient format for search and retrieval. This process is implemented as part of our multi-tiered architecture, focusing on creating inverted files that can be easily consumed by various search engine implementations. Our indexing process takes the cleaned and enriched content from the preprocessing stage and converts it into an inverted file structure, fundamental to efficient information retrieval. Implemented as a Spark batch job, the indexing process

[10] https://resiliparse.chatnoir.eu
[11] https://opencode.it4i.eu/openwebsearcheu-public/spark-deployment
[12] https://www.tira.io/tirex

runs across our HPC infrastructure, allowing for parallel processing of large volumes of data. We partition the index into daily "index-shards" based on metadata derived during preprocessing, such as topic and language, enabling the creation of semantically coherent subsets of the full web index.

This approach to index creation offers significant flexibility. By leveraging various metadata types, we can create specialized index shards. For instance, language-based shards can support country-specific search engines, while topic-based shards can focus on specific areas like news or sports. The indexer produces Common Index File Format (CIFF) files, a standardized format that ensures compatibility with a wide range of open-source search engines.

CIFF[13], a Protobuf schema, describes inverted files in a structured, consistent, and minimal format. A CIFF file contains a header with basic collection statistics, term records including document/collection frequencies and "postings" (i.e. in which document each term occurs), and document records with identifiers and lengths. This standard provides the essential information needed to build a successful search engine, facilitating easy import and transformation of data into various search engine architectures.

3.3. Cluster Deployment Strategy

Our cluster deployment strategy is critical for managing the vast scale of data processed. The indexing process is deployed as an Apache Spark job within our multi-tier cluster setup, optimized for the specific demands of index creation. We utilize tools like Magpie to automate the deployment process, aligning with the scheduling systems of various clusters (e.g., SLURM) to ensure efficient resource allocation and job management.

The data flow in our system is designed for efficiency and scalability. Post-indexing, the CIFF files are stored in our federated iRODS storage system, ready for distribution and use by various search engine implementations. Our current deployment spans multiple HPC centers, including LRZ, IT4I and CSC, with ongoing efforts to scale up the indexer to handle the continuously growing volume of crawled content. Data access requires authentication via the European EUDat / B2ACCESS service [16] which has been integrated with the data center's access and authentication systems.

This comprehensive indexing and deployment strategy enables us to efficiently process and index vast amounts of web data, creating a flexible and powerful foundation for the Open Web Index. By leveraging the power of distributed computing and standardized formats, we're able to create a resource that can support a wide range of search and analysis applications, fostering innovation in web search technology.

3.4. IRODS-Storage and the LEXIS Platform

The Integrated Rule-Oriented Data System (iRODS) is an open-source data management middleware that allows the creation of a unified view over different geographically distributed storage systems[14]. It also maintains a rich database that allows the assignment of metadata to single files or directories and corresponding fast querying capabilities. The LEXIS Platform, initiated through a Horizon 2020 project (GA #825532), adopted iRODS as the main component of its Distributed Data Infrastructure (DDI). In addition, LEXIS also indices iRODS metadata in an ElasticSearch engine to ensure fast full-text queries. On top of iRODS, an asynchronous Staging API is deployed at each computing centre to stage data between storage systems and HPC clusters. This mechanism is leveraged in the indexing and pre-processing workflows executed on HPC resources through the LEXIS Platform.

3.5. Dataset-based Publishing of Daily-index Shard

While the crawlers run continuously within the cloud partitions of the involved HPC centers, HPC workflows are executed once per day to process the crawled data from the previous day at each specific HPC center. Consequently, we publish a daily index slice in the form of a dataset, which is made available under a unique UUID-based identifier via iRODS. The metadata for these datasets partially follows the DataCite Metadata Standard 4.5 [5], containing information about creators, publishers, titles, and license information.

[13] https://github.com/osirrc/ciff

The data is partitioned using a HIVE-like access path scheme[15]: `year=<year>/month=<month>/day=<day>/language=<lang>`. We chose HIVE partitioning because it allows the use of standard tools and simplifies file-based merging of datasets. Additionally, every dataset includes a `changelog.json` file at the root, which indicates any changes made. This feature is necessary to log dataset modifications, such as those required for legal reasons when an external entity requests the removal of certain data items for privacy concerns.

Publishing index shards as daily datasets, instead of updating a single web index daily, offers several advantages:

1. The index is broken down into smaller, self-contained units, typically ranging around 20 GB.
2. Access to individual increments is based on time and metadata, making management easier.
3. Metadata is associated with each dataset, facilitating organization and querying.
4. Complex workloads can be partitioned on a per-dataset basis.
5. Data usage can be tracked per dataset, which is particularly useful when monitoring training data for use in generative AI systems.

However, managing numerous datasets can become complex. To address this, we developed an open-source command-line dataset management tool, *owilix*[14], built on top of LEXIS and B2ACCESS[15]. This tool offers several key features:

1. **Pulling Data:** The primary function of *owilix* is to pull data from remote systems using a specifier concept—essentially, a short query string that describes the data center, time range, and metadata. For example, the command `all:latest/license=OWIV1` pulls the latest datasets from all data centers that have the OWIV1 license. Data is transferred file-based, allowing for additional file filters to limit the data amount.
2. **Pushing Data:** To encourage collaboration in data cleaning, enrichment, and provision, *owilix* also allows users to modify pulled data and subsequently push those modifications back to the server. This pushing process is file-based, enabling, for example, the addition of annotations to web content or the integration of additional indices, such as dense embeddings for Retrieval Augmented Generation or probabilistic indices like Bloom filters.
3. **SQL Queries:** While push and pull operations are file-based, allowing selection based on pre-partitioned data, *owilix* also supports running SQL queries against datasets. This capability is powered by DuckDB[16], which efficiently queries Parquet files on local and remote filesystems. With DuckDB's predicate pushdown and query optimization strategy for Parquet files[13], combined with iRODS-mounted datasets, we can efficiently select rows and columns, reducing data transmission between server and client. The use of the parquet format follows best-practices in big data setup, which have shown to deliver good query performance[7].

Since *owilix* allows the creation of new datasets from existing ones, such as through SQL queries, we also implemented a mechanism to track the provenance of datasets. This is done using a URI-based provenance schema, which references the original dataset and specifies any transformations applied. For example, the URI `owi://UUID/select=*&where="url like de"` links to a dataset with the specified UUID, filtered using the indicated `SELECT` and `WHERE` statements.

Overall, our dataset-oriented approach reduces the complexity of managing large web datasets and provides powerful tools for users to slice and dice the data as needed, making web data manageable even in low-resource settings. An initial URI-based provenance schema allows to track dataset deviates and allows to establish collaborative data curation workflows.

[14] https://opencode.it4i.eu/openwebsearcheu-public/owi-cli
[15] https://b2access.eudat.eu/
[16] https://duckdb.org/

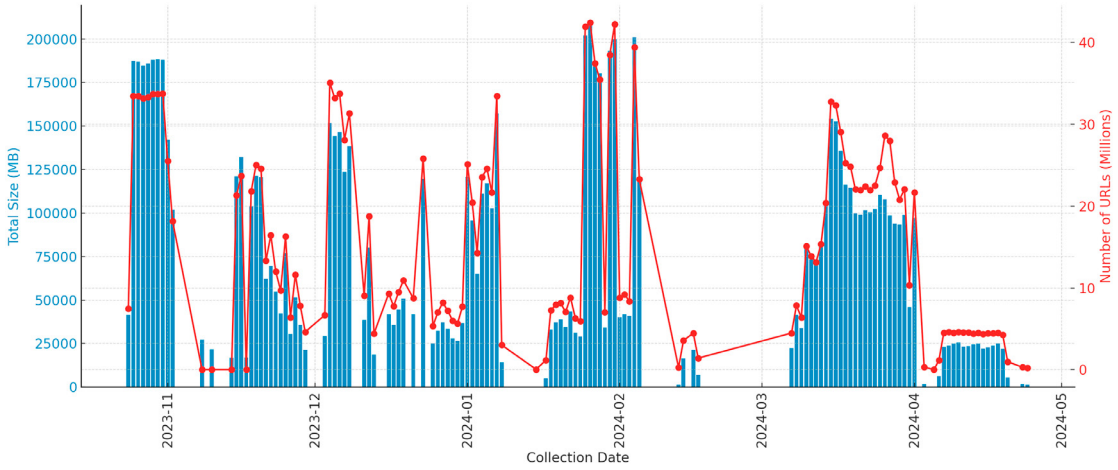

Fig. 3. Dataset size (in MB) and number of million URLs per dataset between October 2023 and April 2024.

4. Results

In this section, we present the results of the first development phase (alpha phase) in terms of data volume and HPC resource utilization. The alpha phase, which began on October 23, 2023, and continued until April 30, 2024, provided valuable insights into the processing and storage capacities required, even though datasets were not generated daily due to ongoing pipeline development. More recent data can be accessed through our online dashboard at https://openwebindex.eu.

4.1. Development and Deployment Status

During the first phase, we had the system running at two data centers: The EuroHPC JU partner IT4Innovations National Supercomputing Center (IT4I) in Ostrava and the Leibniz Supercomputing Center (LRZ) in Munich. Both sites had crawlers running ingesting data to S3 and running the above mentioned preprocessing and indexing pipeline.

After the successful first proof-of-concept, we have extended the setup to five data centers running different components.

The iRODS-based federated data storage is currently deployed at IT4I, LRZ, the German Aerospace Center (DLR) and CSC – IT Center for Science. The iRODS federation has been partially established between the different centers allowing users to access the available data from all the sites available. The access to the data is continually improved based on feedback from both project partners and external users.

The HEAppE middleware that allows the LEXIS orchestrator to access the different HPC resources is deployed at LRZ (Linux Cluster), IT4I (Karolina), and CSC (Puhti cluster). The deployment of HEAppE at DLR is in progress and will allow the LEXIS orchestrator to execute workflows on the newly established Terrabyte infrastructure for geo-spatial data analysis.

4.2. Dataset Size and Index Partitions

For the alpha phase we have crawled approximately 76 TB of raw data on in total 127 different days, which is approximately 500 GB per day. After running the preprocessing and index pipelines, we ended up with in total roughly 2 billion URLs (2,013,843,377) - around 15.9 Million URLs per day - with a cleaned dataset size of ca. 9.2 TB distributed over 172 datasets. Figure 3 shows the dataset distribution in terms of MB and Millions of URLs per Dataset. From the numbers we can derive a raw HTML size of roughly 38 kB and a plain text size of around 5 kB per web page (excluding multimedia elements in both cases). Note that the plain text also contains microformats and hence has some redundancy.

The output datasets are partitioned into daily language-based shards according to ISO-639 Part 3[17]. On average, each daily index slice contains between 300 and 450 shards. The language distribution is skewed significantly, where roughly 40% of the index consists of English documents, and there is a long tail of languages for which we only crawled a few documents. Note that it is as of yet unclear how accurate our language detection module is and how this influences the shard distribution of our datasets. The size of the resulting output datasets (i.e. Parquet files containing cleaned text and extracted metadata combined with the inverted files in CIFF format) is roughly 10-15% of the size of the original (gzipped) WARC files.

4.3. HPC Utilization

For preprocessing and indexing, we usually run our workflows on 4-6 HPC nodes, on which we allocate 12-24 cores depending on the node's available memory. Most of the time (70-90%) is spent on preprocessing and metadata extraction/enrichment, which takes roughly 1-2 minutes per WARC file. For one of the larger datasets (~14k WARC files of 100 MB each), which resulted in an output dataset of ~150 GB the processing times were distributed as follows: preprocessing and enrichment took ~3.5 hours; indexing finished in ~1.5 hours; and copying the data to the iRODS-based DDI took an additional ~1.5 hours.

When datasets grow larger, we can increase the horizontal scaling factor by assigning more nodes to each HPC job in order to ensure index shards can still be produced daily. In case of insufficient capacity within a HPC center, we would have to adjust the amount of data to be processed, either by reshuffling crawled data or adjusting crawling capacities for this particular data center. Nevertheless, our distributed approach allows to scale vertically per HPC data center as well as horizontally accross data centers.

5. Conclusion

In this paper we have presented the utilisation of EuroHPC JU resources for creating an open index for web search, the so-called Open Web Index. Scaling up the necessary storage and compute requires a collaborative effort between HPC centers. To execute workflows across data-centers and for managing the related dataset we have presented the LEXIS platform and our iRODS-based federated data storage. We also developed `owilix` as dataset management tool on top of the federated storage, which allow pushing, pulling and querying of data. In the first development phase lasting from October 23 to April 24 we processed 76 TB of web data which equates to approximately 2 billion URLs. HPC workflows take ~6.5 hours per day and data-center, utilizing an modes amount of 4-6 HPC nodes.

After this successful alpha phase, we significantly increased the data ingestion from 0.5 TB / day up to 3 TB /day with an aim to reach 5 TB / day. This would increase the number of URLs per day by a factor of 6-10, reaching around 95 million to 159 million URLs per day. Furthermore, we increased the up time of the crawler significantly so that we can deliver these numbers on a daily basis. We also assume that compute resources will increase linearly by a factor of 10 yielding to significant compute load for HPC centers. This would again increase when utilizing generative AI based dense indexing techniques, which utilize large language models. However, we hope that in the next 6 months we can provide stable services, delivering daily index patches of 100 million to 200 million URLs per day and thus support innovators and research in need for web data.

Acknowledgements

This work is part of the OpenWebSearch.eu project. The OpenWebSearch.eu Project is funded by the EU under the GA 101070014 and we thank the EU for their support. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). We acknowledge EuroHPC Joint Undertaking for awarding us access to Karolina at IT4Innovations, Czech Republic and LUMI at CSC, Finland.

[17] https://iso639-3.sil.org/

References

[1] Bevendorff, J., Gupta, S., Kiesel, J., Stein, B., 2023. An empirical comparison of web content extraction algorithms, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2594–2603.

[2] Dinzinger, M., Al-Maamari, M., Zerhoudi, S., Istaiti, M., Mitrović, J., Granitzer, M., . OWler: Preliminary results for building a collaborative open web crawler, in: Open Search Symposium 2023, 4-6 October 2023, CERN, Geneva, Switzerland. URL: https://zenodo.org/records/10581841, doi:10.5281/zenodo.10581841. publisher: Zenodo.

[3] Golasowski, M., Martinovič, J., Křenek, J., Slaninová, K., Levrier, M., Harsh, P., Derquennes, M., Donnat, F., Terzo, O., 2022. The lexis platform for distributed workflow execution and data management, in: HPC, Big Data, and AI Convergence Towards Exascale. Taylor & Francis.

[4] Granitzer, M., Voigt, S., Fathima, N.A., Golasowski, M., Guetl, C., Hecking, T., Hendriksen, G., Hiemstra, D., Martinovi'c, J., Mitrovi'c, J., et al., 2023. Impact and development of an open web index for open web search. Journal of the Association for Information Science and Technology doi:10.1002/asi.24818.

[5] Group, D.M.W., 2024. Datacite metadata schema for the publication and citation of research data and other research outputs. URL: https://doi.org/10.14454/g8e5-6293.

[6] Hendriksen, G., Dinzinger, M., Farzana, S.M., Fathima, N.A., Fr"obe, M., Schmidt, S., Zerhoudi, S., Granitzer, M., Hagen, M., Hiemstra, D., et al., 2024. The open web index: Crawling and indexing the web for public use, in: European Conference on Information Retrieval, Springer. pp. 130–143.

[7] Ivanov, T., Pergolesi, M., 2020. The impact of columnar file formats on sql-on-hadoop engine performance: A study on orc and parquet. Concurrency and Computation: Practice and Experience 32, e5523.

[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems 33, 9459–9474.

[9] Lin, J., Mackenzie, J., Kamphuis, C., Macdonald, C., Mallia, A., Siedlaczek, M., Trotman, A., de Vries, A., 2020. Supporting interoperability between open-source search engines with the common index file format, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2149–2152.

[10] Macdonald, C., Tonellotto, N., 2020. Declarative experimentation in information retrieval using pyterrier, in: Proceedings of ICTIR 2020.

[11] Mallia, A., Siedlaczek, M., Mackenzie, J., Suel, T., 2019. Pisa: Performant indexes and search for academia. Proceedings of the Open-Source IR Replicability Challenge .

[12] Qunis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C., 2006. A high performance and scalable information retrieval plateform, in: SIGR workshop on open source information retrieval.

[13] Raasveldt, M., Mühleisen, H., 2019. Duckdb: an embeddable analytical database, in: Proceedings of the 2019 International Conference on Management of Data, pp. 1981–1984.

[14] Rajasekar, A., Moore, R., Hou, C.Y., Lee, C.A., 2010. iRODS primer: integrated rule-oriented data system. Morgan & Claypool Publishers.

[15] Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H., Murthy, R., 2010. Hive-a petabyte scale data warehouse using hadoop, in: 2010 IEEE 26th international conference on data engineering (ICDE 2010), IEEE. pp. 996–1005.

[16] de Witt, S., Lecarpentier, D., van de Sanden, M., Reetz, J., 2017. Eudat-a pan-european perspective on data management, in: 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), IEEE. pp. 1–5.