

Optimization plan



JUNE 2018

Optimization & Upgrade of OpenAIRE Technical
Services

D10.1 – Optimization plan

Version 0.2 – Reviewed

PUBLIC

This deliverable specifies the OpenAIRE Advance data
management plan and
policies.



H2020-EINFRA-2017
Grant Agreement 777541

Document Description

D10.1 – Optimization plan

WP10 - Optimization & Upgrade of OpenAIRE Technical Services	
WP participating organizations: CNR, ICM, CERN, ARC, UNIBI	
Contractual Delivery Date: 06/2018	Actual Delivery Date: MM/YYYY
Nature: Report	Version: 0.2 First Delivery
Public Deliverable	

Preparation Slip

	Name	Organisation	Date
From	Claudio Atzori, Paolo Manghi	CNR	18/06/2018
Edited by	Alessia Bardi	CNR	25/06/2018
Reviewed by	Marek Horst	ICM	02/07/2018
Approved by			
For delivery	Mike Chatzopoulos	UoA	04/07/2018

Revision History

Issue	Item	Reason for Change	Author	Organization
V0.1	Draft version	First draft of deliverable	Claudio Atzori	CNR
V0.2	First Delivery	Review	Marek Horst	ICM
V1.0	Final version			

Table of Contents

TABLE OF CONTENTS	2
1 INTRODUCTION	6
1.1 INFRASTRUCTURE OVERVIEW	6
2 METHODOLOGY	7
2.1 WHITE BOX MONITORING	7
2.2 RESOURCE RATIONALIZATION AND TECHNOLOGY CONSOLIDATION	11
2.3 CONTINUITY OF SYSTEM OPERATIONS.....	12
3 OPTIMIZATION PLAN OVERVIEW	13
3.1 SHORT TERM GOALS – M12.....	13
3.2 MID TERM GOALS - M24	13
3.3 LONG TERM GOALS - M36	14

This document contains description of the OpenAIRE-Advance project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the OpenAIRE-Advance consortium and can in no way be taken to reflect the views of the European Union.

OpenAIRE-Advance is a project funded by the European Union (Grant Agreement No 777541).



Acronyms

CAP	Content Acquisition Policies
API	Application Programming Interface
IIS	Information Inference System
HBASE	Hadoop Database
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
REST	Representational State Transfer
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
WP	Work Package

Publishable Summary

This document describes the action plan for the optimizations and upgrades to be introduced in the OpenAIRE technical services in OpenAIRE Advance. WP10 focuses on activities regarding the optimization and upgrade of the current OpenAIRE TRL8 / TRL9 services in order to further improve the offer to OpenAIRE consumers, be them humans using the portal or third-party services using the public API (api.openaire.eu), as well as to better support the work of OpenAIRE data curators and infrastructure operators. Activities include: performance and scalability optimization, workflow refactoring, functionality refinement, and tailoring of OpenAIRE products in order to meet requirements and feedbacks gathered during the operation of the infrastructure.

1 | INTRODUCTION

1.1 Infrastructure overview

The OpenAIRE information space aggregates about 30 million publications from more than 1,000 data providers. The changes that will be soon introduced in the content acquisition policies (CAP) lay the foundation for a significant increase in the number of bibliographic records describing publications, datasets, software and other research products aggregated and processed by the OpenAIRE services. To this aim the optimizations addressed in WP10 involve several of the subsystems illustrated in figure 1, from the services and components involved in the content acquisition processes, to the architectural solutions adopted for content storage, to the data processing pipelines implementing the OpenAIRE graph population, deduplication, and indexing.

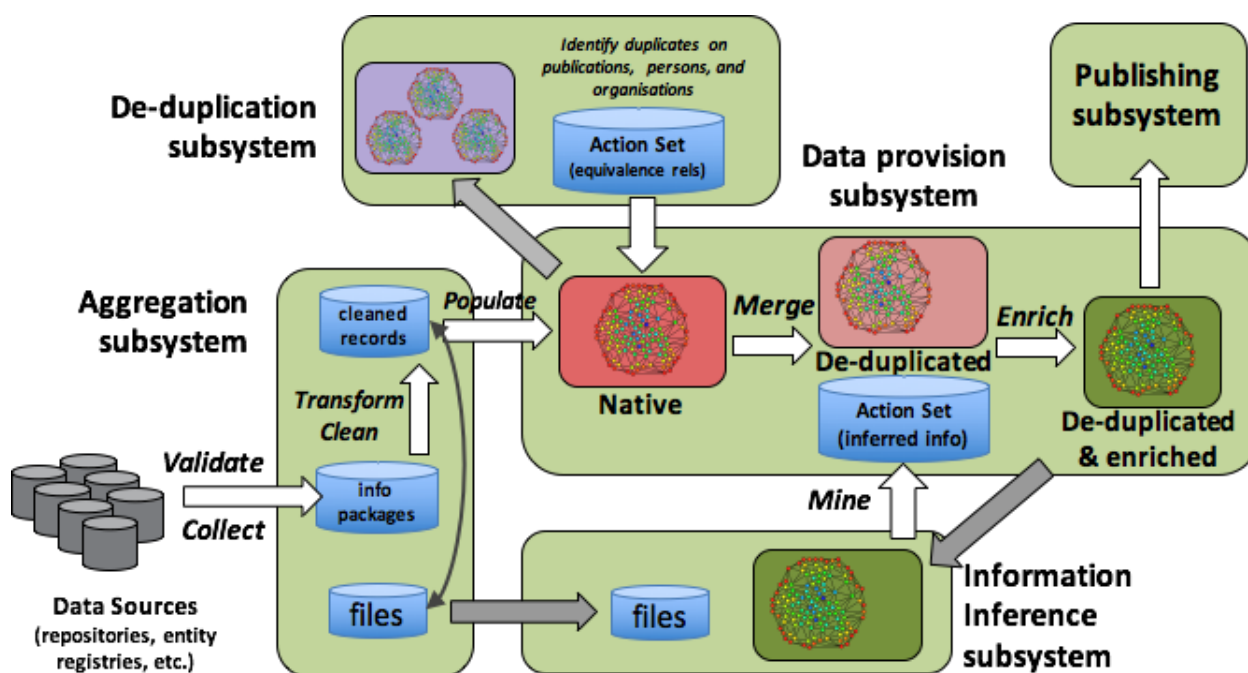


FIGURE 1. ARCHITECTURE OVERVIEW

The optimization actions on the different subsystem will therefore address aspects of processing capability (i.e. throughput) aimed to (i) better distribute the execution of the aggregation workflows and (ii) reduce the time needed to process the data, therefore better supporting the work of data curators and infrastructure operators. Moreover, aspects of system robustness and high availability will be taken into account as so as to improve the system reliability.

2 | METHODOLOGY

Engineering solutions for the system architecture capable to cope with the increasing data volume has become a more and more challenging task, together with the efforts needed to guarantee reliability in the growing ecosystem of services offered by the OpenAIRE technological infrastructure. The methodology will include three main action points:

1. White box monitoring. The pervasive adoption of measures for monitoring systems and applications of the OpenAIRE infrastructure has become a critical aspect to provide reliable services. WP10 will promote and coordinate actions for the adoption and implementation of white box monitoring solutions for the OpenAIRE services by the different teams of developers and system engineers.
2. Resource usage rationalisation and technology consolidation. Task 10.1 will address architectural and implementation aspects characterising the workflows used to produce the “big data graph” of OpenAIRE. The current system in fact is composed of independent, loosely coupled subsystems, based on different technologies and characterised data flows that involves growing data volumes. This calls for a rationalisation of resources allocated on heterogeneous technologies.
3. Continuity of system operations. Such large architectural interventions will take time to be realised and become operational. Meanwhile existing system must continue to support the regular operations carried out by OpenAIRE infrastructure operators (aggregation, deduplication, mining, data provision) therefore, along with the design of the architectural changes described above, WP10 will coordinate an activity aimed to identify critical points of intervention in the current architecture to improve the service performances and address the upcoming data increase.

2.1 White box monitoring

In general, systems tend to be measured at three levels: network, machine, and application. Typically, application metrics are the hardest, yet most important, of the three. They are very specific, tightly bound the business logic and they change as the application (and its requirements) change. The white-box monitoring solutions will be added to the black-box monitoring system (based on [monit](https://mmonit.com)¹) already in place to identify and possibly correct symptoms of malfunctioning in the OpenAIRE infrastructure.

White-box monitoring, contrary to the black box paradigm, depends on the ability to inspect the internal state of the system, such as logs or HTTP endpoints, with instrumentation. White-box monitoring therefore allows detection of imminent problems, failures masked by retries (and so forth), providing important raw input for business analytics, facilitating analysis of security breaches, breaking down the observations among the different software layers involved in the implementation of a certain functionality (therefore leveraging on the responsibilities among the development teams), providing input for the capacity planning of the different levels of a multi-

¹ <https://mmonit.com>

layered system. Overall, system (and application) monitoring is not a novel concept, but it plays a crucial role in the system design, enabling for long-term trends analysis, comparing over time or experiment groups, alerting, building dashboards, conducting ad hoc retrospective analysis (i.e. debugging). Good monitoring system should enable infrastructure operators to address issues by answering two questions: what's broken, and why? The "what's broken" indicates the symptom; the "why" indicates a (possibly intermediate) cause. "What" versus "why" is one of the most important distinctions in designing good monitoring with maximum signal and minimum noise². To implement the concept of white box monitoring in OpenAIRE the tool of choice is Prometheus³. Prometheus is an open source white box monitoring solution that uses a time-series database to provide scraping, querying, graphing and alerting based on time-series data. For the definition of the dashboards instead, the plan is to use Grafana⁴, an open source, feature rich metrics dashboard and graph editor. In the following table we provide an overview of alternative monitoring tools and their comparison with Prometheus. For the choice we considered the following criteria, whose evaluation is summarised in the column *notes*:

- Being Open Source
- Free availability of all the functionalities
- Ease of integration in the OpenAIRE services ecosystem
- Adherence to the use case

	Open Source / Free availability of all the functionalities	Notes
Graphite⁵	Yes / Yes	Prometheus offers a richer data model and query language, in addition to being easier to run and integrate into the OpenAIRE environment
InfluxDB⁶	Yes / Some advanced features are available only in the commercial version	More oriented to event logging applications. Prometheus is better suited to build dashboards on top of metrics, in addition to providing better alerting, and notification functionality.

² <https://landing.google.com/sre/book/chapters/monitoring-distributed-systems.html>

³ <https://prometheus.io>

⁴ <https://grafana.com>

⁵ <https://graphiteapp.org>

⁶ <https://www.influxdata.com>

OpenTSDB⁷	Yes / Yes	Although its being backed on Hadoop HBase, which could make it be an interesting choice, OpenTSDB lacks a full query language, only allowing simple aggregation and math via its API.
Nagios⁸ or Sensu⁹	Yes / Yes (for both)	Nagios and Sensu are suitable for basic monitoring of small and/or static systems where blackbox probing is sufficient.
ELK¹⁰	Yes / Some advanced features are available only in the commercial version	The ELK (Elasticsearch / Logstash / Kibana) stack is better suited to aggregate, analysis and graphing application logs.

⁷ <http://opentsdb.net>

⁸ <https://www.nagios.org>

⁹ <https://sensu.io>

¹⁰ <https://www.elastic.co/elk-stack>

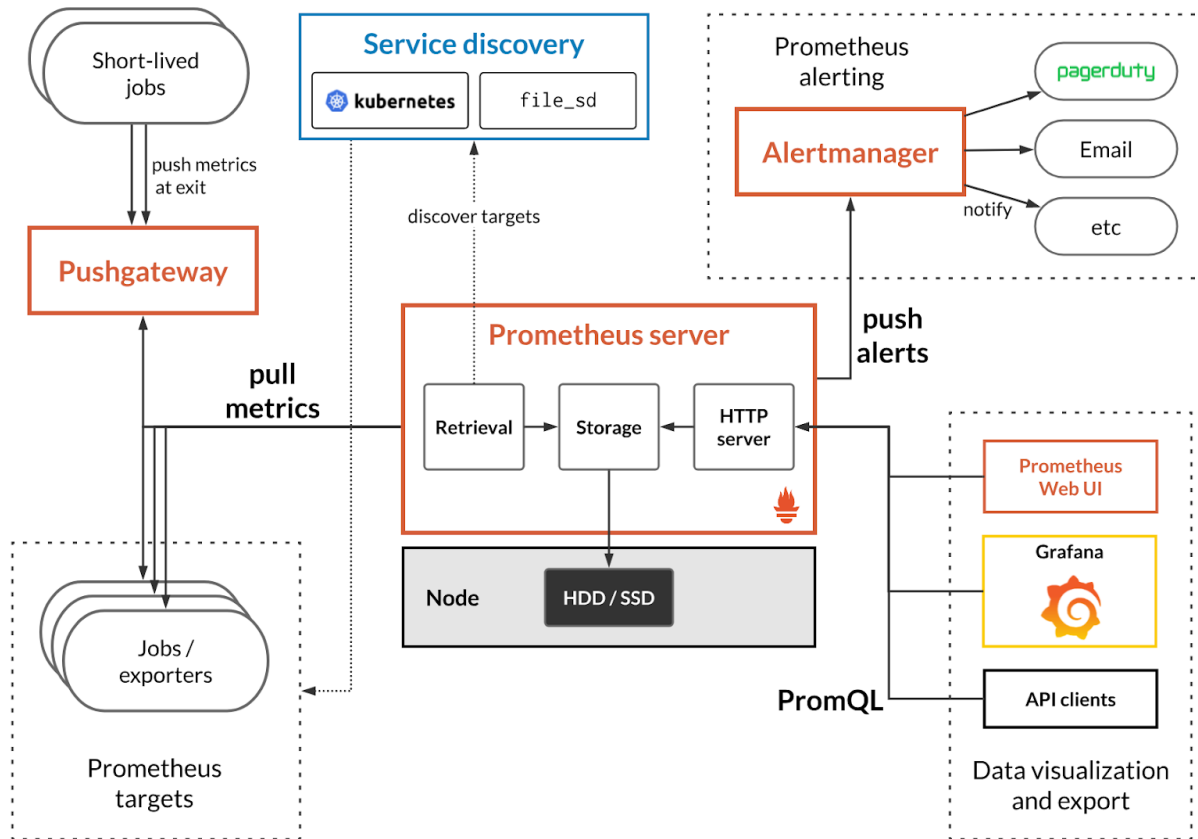


FIGURE 2 PROMETHEUS ARCHITECTURE¹¹

Thanks to the insights provided by the monitoring dashboards it will be possible to adjust the resources allocated to implement a given functionality in response to changes in its requirements or in the data. This will facilitate the overall workflow of system design, development, operation and maintenance: assuming that the monitoring system can catch the impact of every change introduced in the production system, updates can be delivered more frequently. Moreover, the OpenAIRE context has been often characterized by a constant evolution of the requirements driving the technical activities, for example the upcoming changes in the CAP will have significant impact on the resources needed to store and process the bibliographic records and the relative full text files. In this context the possibility to prototype a solution and observe its behaviour and impact on the available resources provides an important advantage. To provide an exhaustive view of the waterfall effects solely on the Solr server¹² backing the OpenAIRE search functionalities we suggest to read the in-depths on the technical article from Lucidworks: [“Sizing Hardware in the Abstract: Why We Don’t Have a Definitive Answer”](#).

¹¹ Source: <https://prometheus.io/assets/architecture.svg>

¹² <http://lucene.apache.org/solr>

The principles discussed so far can be tied together into a philosophy on monitoring and alerting that, albeit a bit aspirational, represents a good starting point for defining new alerts providing useful insights on the behaviour of the system. Therefore, among the objectives of WP10, we foresee a goal that goes beyond the optimization plan described in this document, as it can help OpenAIRE technical and management teams ask the right questions, regardless of the size or the complexity of the system. Activities for the introduction of the monitoring dashboards will tackle individual subsystems and will last for the entire project duration. Priorities will be assigned in order to address first the most critical services, but at this stage it is not possible to draft a detailed action plan.

2.2 Resource rationalization and technology consolidation

Task 10.1 will address architectural and implementation aspects characterising the workflows used to produce the “big data graph” of OpenAIRE. Architectural aspects involve the structure of components and subsystems, their interrelationships, communication patterns, and the principles and guidelines driving their design and evolution over time. Implementation aspects instead derive from the technologies used to address the realisation of the system functionalities. For example, in figure 1 the aggregation subsystem can be seen as a distinct application, in fact it is based on the D-Net framework¹³, an Open Source service-oriented solution for the construction of customized aggregative data infrastructures. D-Net is based on its own architecture, where a manager service orchestrates the data aggregation workflows. Other subsystems in the architecture diagram are instead based on technologies from the Hadoop ecosystem. This implies that the data synchronization mechanisms among the different subsystem will pay an increasing cost in response to increasing data volumes produced upstream. Moreover, as the Hadoop ecosystem provide tools for scalable data persistence and processing, it is natural to think about centralising part of the functionalities, implementing them within the same Hadoop technology. This rationalisation overall will generally provide benefits on different perspectives:

- Data locality: the data will be available for processing without needing to be transferred from one system to another;
- Maintenance cost: less technological fragmentation implies that system engineers will have to maintain less different technologies;
- Better use of physical resources and scalability: physical resources allocated to the Hadoop ecosystem will contribute to every single part of the system implemented on top of it, moreover the rationalisation of the system workflows onto a single Hadoop cluster will provide a more uniform and optimal use of its resources.

This general idea is quite aspirational, as clearly not every OpenAIRE functionality can be implemented on top of a Hadoop based technologies. In fact, Hadoop works well on batch processing use cases rather than on tasks that require real time analytics. For example, the search and browse functionality implemented on the OpenAIRE portal, already implemented on

¹³ <http://www.d-net.research-infrastructures.eu>

top of Apache Solr, imposes strict requirements for the time needed to serve a query (in the order of milliseconds). Instead the aggregation workflows responsible to collect and process thousands of potentially large batches of bibliographic metadata records are good candidates to be implemented on top of Hadoop.

For this reason, the first activity to be addressed is a feasibility study aimed to identify which, among the functionalities implemented across rather sparse subsystems, are well suited to be implemented on top of Hadoop and which needs to be implemented on different technologies better supporting the specific use case.

2.3 Continuity of system operations

The architectural changes to be introduced for the resource usage rationalisation and technology consolidation interventions will take time to be realised and to become fully operational, in this timeframe the OpenAIRE engineers still need to address the evolution of the system requirements deriving from the data growth as so as to ensure continuity in the workflow executions. Clearly invasive architectural changes require considerable amount of work and dedication from the technical partners, therefore we first categorize the goals based on the expected delivery time as presented in the optimization plan, where we consider as short-term M12, mid-term M24, long term M36. Moreover, a preliminary analysis of the OpenAIRE subsystems and components has been conducted to identify points of interventions that would allow to mitigate issues that could arise during the transition period. Such analysis identified two major points of action that will be further discussed in section 3:

- A. Extend the support for incremental aggregation and data provision;
- B. Improvement of the Solr based search functionalities.

3 | OPTIMIZATION PLAN OVERVIEW

3.1 Short term goals – M12

- D. Extend the support for incremental aggregation and data provision: incremental aggregation aims at reducing the amount of data moved on the wire and processed by the aggregator. In order for the OpenAIRE aggregation subsystem to support incremental aggregation the data provider must support it in the first place, i.e. it must be able to provide the records changed after a given moment in time; in this way the aggregation in incremental mode will only updates those records, therefore reducing the usage of computational resources, especially for large data providers. Complementarily third party clients willing to consume OpenAIRE's information space from the oai-pmh endpoint¹⁴ will be able to work in incremental mode, i.e. to consume only those records that changed on the original source after a given moment in time.
- E. Introduction of monitoring dashboards for the public services (API, portal).
- F. Revision of the resources allocated on different services according to the feedback provided by the monitoring dashboards.

3.2 Mid term goals - M24

- G. Migration towards the OCEAN data centre¹⁵: the data centre that hosted OpenAIRE technical services for years is lacking spare resources to address future extensions, therefore in tight collaboration with ICM, the technical services will be gradually migrated from the current server farm to a different one. This activity will provide benefits on several fronts:
 - Better support from hardware providers, which imply better mitigation of potential failures;
 - More powerful hardware and network connectivity supporting better performances;
 - Part of the recently introduced services are already hosted in the OCEAN data centre, therefore following the data and service locality principle it is important to migrate also those still running in the old server farm;
 - Better energy efficiency.
- E. Introduction of S3 compatible content storage: full text files will be moved from the current NFS solution to an S3 compatible backend and the object storage services adjusted accordingly to support the different technology.
- F. Improvement of the Solr based search functionalities: The OpenAIRE search functionality is implemented on top of Apache Solr (currently at version 4.9.0). A new cluster of servers

¹⁴ http://api.openaire.eu/oai_pmh

¹⁵ <http://ocean.icm.edu.pl/en>

running the newer version 7.3.1 has been already deployed and is being tested to support search operations over the growing information space. The testing methodology is a combination of read and write operations that mimic the usage pattern of the production system environment, where the number of indexed documents can be pushed to hundreds of millions, as well as the query pressure can be adjusted to simulate peaks of requests. The effects of the testing sessions are observed on the Grafana dashboard to assess the optimal configuration settings and resource allocation.

- G. Revision of the resources allocated on different services according to the feedback provided by the monitoring dashboards

3.3 Long term goals - M36

- G. Rationalisation of the the data flows

- a. Merge the two OpenAIRE clusters: Currently the OpenAIRE infrastructure relies on two distinct Hadoop clusters: one dedicated to the information space population, deduplication, and provision, the other dedicated to the information inference subsystem (IIS). The consolidation of such activities on one single Hadoop cluster will provide benefits on different fronts: (i) resource usage, (ii) management efforts, (iii) data flow optimizations.

- b. Design and implementation of the aggregation services as Hadoop jobs: this study is already ongoing and so far has identified the following

- Metadata harvesting, transformation, validation workflows;
- Calculation of information space statistics: the current solution is based on PostgreSQL relational database, alternative tools part of the Hadoop ecosystem are being evaluated (Hive¹⁶, Impala¹⁷).

- H. Introduce support for incremental deduplication: The deduplication process scans selected entity types in the entire Information space to identify duplicated objects. This process currently doesn't exploit any previous knowledge from previous executions. This activity will design mechanisms to define a Ground Truth out of validated deduplication results, to be reused in subsequent scans of the information space as so as to reduce the complexity of the operation.

- I. Revision of the resources allocated on different services according to the feedback provided by the monitoring dashboards.

¹⁶ <https://hive.apache.org>

¹⁷ <https://impala.apache.org>