Hewlett Packard Labs

DHBW Duale Hochschule Baden-Württemberg

Development and demonstration of a proof-of-concept for the integration of programming frameworks for high performance computing into a container-based workflow orchestrator.

2. Project Paper

submitted on October 9, 2023

Faculty of Economics

Degree program in Business Informatics

International Management for Business and Information Technology

von

JON ECKERTH

Practice supervisor :	DHBW Stuttgart:
⟨ Hewlett Packard Labs ⟩	⟨ Titel, Vorname und Nachname ⟩
⟨ Harumi Kuno ⟩	⟨ der/des wissenschaftlichen Betreuerin/Prüferin (TBD) ⟩
⟨ Principal Research Scientist ⟩	

Signature of the practice supervisor:

Author's declaration

Hereby I solemnly declare:

1. that this 2. Project Paper titled *"Development and demonstration of a proof-of-concept for the integration of programming frameworks for high performance computing into a container-based workflow orchestrator."* is entirely the product of my own scholarly work, unless otherwise indicated in the text or references, or acknowledged below;

2. I have indicated the thoughts adopted directly or indirectly from other sources at the appropriate places within the document;

3. this 2. Project Paper has not been submitted either in whole or part, for a degree at this or any other university or institution;

4. I have not published this 2. Project Paper in the past;

5. the printed version is equivalent to the submitted electronic one.

I am aware that a dishonest declaration will entail legal consequences.

Dublin, October 9, 2023

Jon Eckerth

Abstract:

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

List of abbreviations

AI	Artificial Intelligence
k8s	Kubernetes
ML	Machine Learning
HPC	High Performance Computing
CC	Cloud Computing
CI/CD	Continuous Integration/Continuous Delivery
IaC	Infrastructure as Code
PoC	Proof of Concept
HPE	Hewlett Packard Enterprise
CNCF	Cloud Native Computing Foundation
CLASP	Cloud Application Services Platform
CWL	Common Workflow Language
SME	Subject Matter Expert
TCPP	Tightly Coupled Parallel Problem

List of Figures

List of Tables

1 Introduction

In this section, the underlying motivation of this project is explained. Furthermore, the problems which will be addressed by this project are described, which serve as the basis for the research questions which will guide this project and ultimately result in solutions and further questions which are listed in the contributions section and discussed in the conclusion.

1.1 Motivation

The proliferation of "Big Data" has led to the need to compute, analyse, and visualize ever-increasing amounts of datasets, which themselves are getting more and more complex, has led to an ever increasing demand for more efficient and quicker ways to process data.

Both the High Performance Computing (HPC) and the Cloud Computing (CC) community have been working on solutions to distribute and parallelize computations for decades, both with their own approaches and solutions to their respective problems.

While the HPC community has been putting a lot of effort into developing new and extremely efficient ways to parallelize computations, the CC community has been focusing on improving the flexibility, scalability and resilience of their solutions as well as improving the ease of use for their developers and users.

Both used to be very distinct and separate communities due to their very different usecases, while the HPC community was mostly concerned with scientific computing and simulations of physical phenomena, the CC community is mostly concerned with providing a reliable and easily up and down scalable infrastructure for the industry and businesses.

Now with the advent of Machine Learning (ML) and Artificial Intelligence (AI) the two communities are starting to converge, as the ML and AI community is adopting the tools and techniques of both communities to solve their problems as they see fit.

But this convergence of the two is not without its problems, being developed in two coexisting and separate communities, the tools and techniques of both communities are not always compatible with each other, the goal of this project is to find a way to bridge this gap and to find a way to combine the best of both worlds.

1.2 Problem Statement

The following key problems have emerged from the convergence of High Performance Computing (HPC) and Cloud Computing (CC) communities, especially in the context of Machine Learning (ML) and Artificial Intelligence (AI) research:

- **Workload Resilience and Fault Tolerance in HPC:** HPC systems often lack mechanisms to recover from task failures within larger jobs, running for an extended time. When a component task fails, it can invalidate the entire computation, requiring a restart from scratch. This need for resilient failover and verification strategies as well as the need to avoid computational wastage is a key challenge for HPC systems, especially with every increasing system sizes and complexity.[1]

- **Environment/Package Management in HPC:** HPC systems are notorious for their complex package management systems. As having a shared infrastructure between many users each with their own specific needs and requirements of different versions of packages, libaries and software, all the while sharing a common environment. Many solutions to this problem have been developed, each with their own advantages and disadvantages.[2][3][4][5]

- **Portability Issues with HPC:** Tieing in with the previous point, HPC systems are often designed to be optimized for specific hardware as well as having a very specific software stack. This makes the portability of applications between different HPC systems very difficult and often infeasible.[6] This lack of portability contrasts sharply with the more platform-agnostic nature of CC environments, where the containerization of applications has become the norm for ensuring portability.

- **Scalability and Flexibility in HPC:** Due to its direct access to the hardware and very specific hardware needs, HPC systems are often hard to dynamically scale and inflexible. while CC systems are designed to be easily scalable and flexible and are often designed to be hardware agnostic and abstract away the underlying hardware. This becomes especially relevant in the context of heterogeneous hardware, where the hardware is not uniform and consists of different types of hardware, which is becoming more and more common in the context of ML and AI research.

- **Lack of Interconnected Problem Solving in CC:** The workloads traditionally deployed on CC systems are often independent of each other, like load balancing, web hosting, etc. This is in stark contrast to the interconnected nature of HPC workloads, where each part of the input data is dependent on the other parts of the input data, such that all nodes of the system need to be able to communicate with each other.

[1] Egwutuoha et al. 2013
[2] Dubois/Epperly/Kumfert 2003
[3] Bzeznik et al. 2017
[4] Gamblin et al. 2015
[5] Hoste et al. 2012
[6] Canon/Younge 2019, p. 50

- **Provenance and Reproducibility:** Another need that is becoming more and more important in the context of ML and AI research is the need for provenance and reproducibility of results. Being able to tell which data was used to train the model, is of ever increasing importance as the influence the resulting models have on our lives increases as well as the data used to train the model. This is especially important since it is crucial to ensure that the data is not biased, outdated, or otherwise flawed, which could lead to incorrect predictions, decisions, or recommendations. in addition various data sources, from images to text, may have copyright restrictions that, when overlooked, can lead to significant legal complications.

- **Versioning Limitations:** The dynamic nature of ML and AI research necessitates robust versioning solutions for data, configurations and code. While CC has developed many solutions to this problem over the years, making them their own subsection of the ecosystem, namely Continuous Integration/Continuous Delivery (CI/CD) tools for the testing and deployment of applications aswell as Infrastructure as Code (IaC) tools for the deployment of infrastructure. While many solutions have been developed for the one-off deployment of HPC systems, the dynamic nature of CC systems necessitates a more robust solution to this problem, from which the HPC community could benefit aswell.

1.3 Research Questions

To address the aforementioned problems, to bridge the gap between the two paradigms and to combine the best of both worlds, an integration of the two paradigms is needed. This was accomplished by integrating a HPC framework called 'Arkouda[7]' into a container based CC workflow management tool called 'Pachyderm'[8] and integrating both with the supporting infrastructure the CC system enables us to use. This process of integration and prototyping as well as the explanation of the underlying concepts and technologies will be the focus of this project.

- **RQ1:** *How can a high-performance computing framework be effectively integrated into a container-based workflow management tool?*

- **RQ2:** *What are the opportunities for improving the integration of high-performance computing frameworks with container-based workflow management tools?*

[7] Merrill/Reus/Neumann 2019
[8] pachyderm 2023

1.4 Contributions

In order to address the problems stated above, find answers to the research questions and to bridge the gap between the two paradigms, the following contributions were made:

- **C1:** *An analysis of the problem space and existing solution, within the constraints of time, resources and businesses needs.*
- **C2:** *A prototype implementation combining the 'Arkouda' framework with the Kubernetes (k8s) based workflow orchestrator 'Pachyderm'*
- **C3:** *Further integrations of tools from both sides of the spectrum, addressing many of afformentioned painpoints*

2 Methodology

2.1 Prototyping

Needs to have a methodology from the Spectrum of Methodologies for Business information systems[9]

Argumentation why this project is centrally a Prototyping project:

- The research questions are directly inspired bt the needs of the customer - The limitations and the scope are both defined by the available resources of the business unit as well as the time constraints of the project and the available know-how - test

[9] Wilde/Hess w.y.

3 State of the Art

3.1 Containerization

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Container Solutions

Software defined Infrastructure

Large Scale Container Orchestration

3.2 Workflow Management Tools

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Apache Airflow

Pachyderm

Kubeflow

Apache Oozie

3.3 High Performance Computing Frameworks

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Apache Spark

MapReduce

Apache Flink

Dask

3.4 Additional Tooling

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Versioning

CI/CD

Fabric attached Memory

Object Storage

4 Creation of the Artifact

4.1 Initial Goals

As this project was first and foremost a project, designed to interactively explore the problemspace from the perspective of the HPC community, all the while being contained by business requirements and time constraints, the initial goals of this project were very broad and open ended. At first the initial goal was simply to create a Proof of Concept (PoC) of a realistic workflow engine using the "Arkouda" project, in order to present the Customer with a easily graspable example of its capabilities.

While we are approaching the problem from the perspective of the HPC community, the intended enduser of this tool are the data scientists and Subject Matter Expert (SME)s that are working with the HPC systems, and therefore the tool needs to be designed and selected with the fact in mind that the enduser will most likely not be knowledgeable in the field of HPC or the underlying infrastructure.

In the first iteration of the project a preselector of possible Workflow management tools was presented from the business side, with the option to increase the scope if the presented tools were not sufficient.

Therefore the goals of the first iteration of this project was twofold, first to determine which, if any, of the presented tools were suitable for the task at hand, and to determine what would make an adequate PoC for the customer.

4.2 Selection of Workflow Management Tools

As described in the previous section, the first iteration of this project was to determine which, if any, of the presented tools were suitable for the task at hand. The initial choice of tools was:

- **Pachyderm:** A k8s based Workflow manager, written in go which was recently aquired by Hewlett Packard Enterprise (HPE).
- **Argo:** A k8s based Workflow manager , written in go, which is a Cloud Native Computing Foundation (CNCF) project[10].
- **Cloud Application Services Platform (CLASP):** An in-house developed workflow manager, written in Java, utilizing Serverlet to execute workflows[11].

[10] *Argoproj/Argo-Workflows* 2023

[11] Sayers et al. 2015

- **Snaplogic:** A commercial low-code/no-code workflow manager with a focus on data integration and data engineering[12].

But given that it was possible to select projects outside of the initial selection, the following projects also need to be considered:

- **Airflow:** A Python-based workflow manager under the CNCF umbrella, known for its easy-to-use interface and extensibility[13].
- **Kubeflow:** A k8s-native platform for deploying, monitoring, and running ML workflows and experiments, also a CNCF project, streamlining ML operations alongside other Kubernetes resources[14].
- **Knative:** An open-source k8s-based platform to build, deploy, and manage modern serverless workloads, simplifying the process of building cloud-native applications[15].
- **Luigi:** An open-source Python module created by Spotify to build complex pipelines of batch jobs, handling dependency resolution, workflow management, and visualization seamlessly[16].
- **Common Workflow Language (CWL):** An open-standard for describing analysis workflows and tools in a way that makes them portable and scalable across a variety of software and hardware environments, from workstations to cluster, cloud, and high-performance computing environments.

Selection Criteria

Due to this extensive list of diverse tools, a set of criteria was established to determine which tool would be the most suitable for the task at hand.

- **Ease of use:** As the inteded are not primarily HPC experts, the tool needs to be easy to use and understand, and should not require the enduser to have a deep understanding of the underlying infrastructure. While we can expect that the administration of the infrastructure will be done by adequately trained personnel, the enduser should be spared having to adapt to the underlying infrastructure as much as possible.
- **Extensibility:** One significant constraint of the project is the restricted number of available work-hours. Given that the project's environment predominantly centers around HPC (High Performance Computing) workloads, it's essential for the tool to be easily expandable without requiring extensive modifications to the underlying system. Idealy this property

[12] *iPaaS Solution for the Enterprise* 2023
[13] Haines 2022
[14] *Kubeflow* 2023
[15] *Home - Knative* 2023
[16] *Luigi/Doc/Workflows.Rst at 54a347361ae1031f06105eaf30ff88f5ef65b00c · Spotify/Luigi* 2023

would be transfered to the enduser, allowing them to easily extend the developed tool further to their needs.

- **Community, Support and Documentation:** It is not enough that the software technically permits extensibility, the software also needs to be adequately documented and a support framework needs to be in place. Be it a community of users or a dedicated support team, the enduser and the developers need to be able to rely on the software being maintained and updated aswell as being able to find expert help in case of problems.
- **Maturity:** With the boom of AI and ML in recent years[17], the number of tools and frameworks has exploded, and while this is a good thing it also means that a lot of these tools are still paving their way and are developing rapidly. While this is not necessarily a bad thing, it does mean that the tool might not be ready for production use and might not be able to provide the stability and reliability that is required for a production environment or are lacking in documentation and support.
- **Strategic alignment with HPE:** As this project is being developed within the context of HPE, it is important to consider the strategic alignment of the tool with HPE. HPE has is a large company with a diverse portfolio of products and services, and this project intersects with many different parts of the company. Therefore it is important to consider the strategic alignment of the tool with HPE and its products and services.
- **License:** While this PoC is not a commercial product in itself but rather an exploration of the problem space and a demonstration of what a final commercial product might be like, it is important to consider the licenses of the tools that are being used. Having to strip out a tool later on because of licensing issues would be a significant setback and therefore needs to be considered.
- **Cost:** Time is not the only constraint of this project, as the project is being developed within the context of HPE it is important to consider the cost of the tools that are being used.

Evaluation

The following table shows the evaluation of the tools based on the selection criteria:

[17] *24 Top AI Statistics & Trends In 2023 – Forbes Advisor* 2023

4.3 Design of the Artifact

As can be seen in figure 1, the artifact is composed of 3 main components, the **Central Workflow Engine** which is responsible for the orchestration of the workflows (center), the **HPC Cluster** which is responsible for the execution of TCPP workloads (left) and the **Usability and Support Services** which aim at improving the usability and accessability for the enduser (right).

All this is build on top of the **Heydar Cluster** which has been specifically set up for this project and is described in more detail in section 4.4.1.

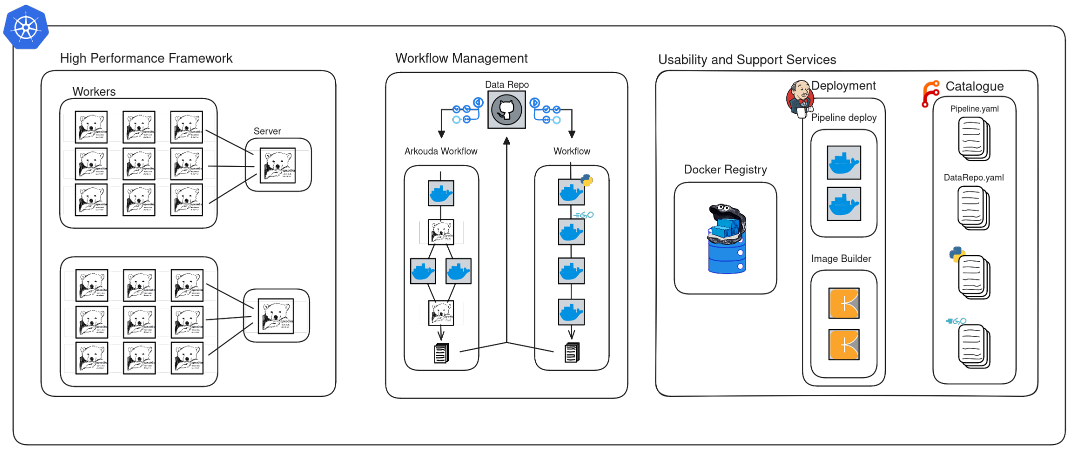

Abb. 1: Pachykouda high level infrastructure diagramm

4.4 Implementation of the Artifact

This section will describe the iterative process of implementing the larger artifact and is broken up into 3 subsections. While these steps where happening concurrently, they each address a different aspect of the project and therefore underwent their own iterative processes.

4.4.1 Infrastructure

Minikube

Heydar Cluster

4.4.2 Usability Impovements

4.4.3 TCPP Workloads

4.5 Evaluation of the Artifact

5 Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6 Summary and Outlook

6.1 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2 Outlook

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

List of literature

24 Top AI Statistics & Trends In 2023 – Forbes Advisor (2023). URL: `https://www.forbes.com/advisor/business/ai-statistics/` (retrieval: 10/07/2023).

Argoproj/Argo-Workflows (2023). Argo Project. URL: `https://github.com/argoproj/argo-workflows` (retrieval: 10/07/2023).

Bzeznik, B./Henriot, O./Reis, V./Richard, O./Tavard, L. (2017): Nix as HPC Package Management System. In: *Proceedings of the Fourth International Workshop on HPC User Support Tools*. HUST'17. New York, NY, USA: Association for Computing Machinery, pp. 1–6. ISBN: 978-1-4503-5130-0. DOI: `10.1145/3152493.3152556`. URL: `https://dl.acm.org/doi/10.1145/3152493.3152556` (retrieval: 10/03/2023).

Canon, R. S./Younge, A. (2019): A Case for Portability and Reproducibility of HPC Containers. In: *2019 IEEE/ACM International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)*. 2019 IEEE/ACM International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC). Denver, CO, USA: IEEE, pp. 49–54. ISBN: 978-1-72816-028-3. DOI: `10.1109/CANOPIE-HPC49598.2019.00012`. URL: `https://ieeexplore.ieee.org/document/8950982/` (retrieval: 10/04/2023).

Dubois, P./Epperly, T./Kumfert, G. (2003): Why Johnny Can't Build [Portable Scientific Software]. In: *Computing in Science & Engineering* 5.5, pp. 83–88. ISSN: 1558-366X. DOI: `10.1109/MCISE.2003.1225867`. URL: `https://ieeexplore.ieee.org/abstract/document/1225867` (retrieval: 10/03/2023).

Egwutuoha, I. P./Levy, D./Selic, B./Chen, S. (2013): A Survey of Fault Tolerance Mechanisms and Checkpoint/Restart Implementations for High Performance Computing Systems. In: *J Supercomput* 65.3, pp. 1302–1326. ISSN: 1573-0484. DOI: `10.1007/s11227-013-0884-0`. URL: `https://doi.org/10.1007/s11227-013-0884-0` (retrieval: 10/03/2023).

Gamblin, T./LeGendre, M./Collette, M. R./Lee, G. L./Moody, A./de Supinski, B. R./Futral, S. (2015): The Spack Package Manager: Bringing Order to HPC Software Chaos. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '15. New York, NY, USA: Association for Computing Machinery, pp. 1–12. ISBN: 978-1-4503-3723-6. DOI: `10.1145/2807591.2807623`. URL: `https://dl.acm.org/doi/10.1145/2807591.2807623` (retrieval: 10/03/2023).

Haines, S. (2022): 'Workflow Orchestration with Apache Airflow'. In: *Modern Data Engineering with Apache Spark: A Hands-On Guide for Building Mission-Critical Streaming Applications*. Springer, pp. 255–295.

Home - Knative (2023). URL: `https://knative.dev/docs/` (retrieval: 10/07/2023).

Hoste, K./Timmerman, J./Georges, A./De Weirdt, S. (2012): EasyBuild: Building Software with Ease. In: *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*. 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, pp. 572–582. DOI: `10.1109/SC.Companion.2012.81`. URL: `https://ieeexplore.ieee.org/abstract/document/6495863` (retrieval: 10/03/2023).

iPaaS Solution for the Enterprise (2023). SnapLogic. URL: `https://www.snaplogic.com/` (retrieval: 10/07/2023).

Kubeflow (2023). Kubeflow. URL: `https://www.kubeflow.org/` (retrieval: 10/07/2023).

Luigi/Doc/Workflows.Rst at 54a347361ae1031f06105eaf30ff88f5ef65b00c · Spotify/Luigi (2023). GitHub. URL: `https://github.com/spotify/luigi/blob/54a347361ae1031f06105eaf30ff88f5ef65b` `doc/workflows.rst` (retrieval: 10/07/2023).

Merrill, M./Reus, W./Neumann, T. (2019): Arkouda: Interactive Data Exploration Backed by Chapel. In: *Proceedings of the ACM SIGPLAN 6th on Chapel Implementers and Users Workshop*. CHIUW 2019. New York, NY, USA: Association for Computing Machinery, p. 28. ISBN: 978-1-4503-6800-1. DOI: `10.1145/3329722.3330148`. URL: `https://dl.acm.org/doi/10.1145/3329722.3330148` (retrieval: 10/04/2023).

pachyderm (2023): Pachyderm. Github repository. URL: `https://github.com/pachyderm/pachyderm` (retrieval: 10/04/2023).

Sayers, C./Laffitte, H./Reddy, P./Ozonat, K./Sayal, M./Simitsis, A./Singhal, S./ Koutrika, G./Das, M./Aji, A./Bosamiya, H./Riss, M./Wilkinson, K./Lucio, J./ Cantal, A./Carvalho, C. (2015): Cloud Application Services Platform (CLASP): User Guide, Introduction, and Operation. In: pp. 1–60.

Wilde, T./Hess, T. (w.y.): Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung. In.