FORCE on Snakemake
Bachelor Thesis Exposé

14.01.2023
Luis Neuhaus

1. Einleitung

Als Workflows werden die Koordination, Ausführung und Dokumentation unabhängiger Programmschritte bezeichnet, die bereits bestehende Daten analysieren oder Daten generieren [1]. Dabei teilt sich die gesamte Berechnung in eine Abfolge von einzelnen Prozessen für meist wissenschaftliche Analysen auf, deren einzelne Zwischenergebnisse als Input für den nächsten Prozess fungieren können. Die Summe dieser Abhängigkeiten können als azyklischer gerichteter Graph repräsentiert werden [2]. Die Kanten stellen dabei den Datenfluss und die Knoten die einzelnen Prozesse dar, die je nach Abhängigkeit parallel oder nacheinander abgearbeitet werden können [2]. Meistens sind solche Workflows für eine bestimmte Aufgabe geschrieben und funktionieren nur in der speziellen Systemumgebung, für die sie erstellt wurden.
Bei umfangreichen Analysen wird mit großen Datensätzen gearbeitet, z.B.: in der Bioinformatik (Genomanalyse [2] oder das Überwachen von Klima- und Vegetationsveränderungen durch die Analyse von Satellitenbildern [3]). Hier kommen (Workflow Management Systeme) WMS zum Einsatz, die Workflows reproduzierbar und skalierbar machen sollen und diese in nicht vollständig spezifizierten Settings ausführen können [3]. Unter anderem bieten sie Möglichkeiten zur Parallelisierung, die besonders wichtig bei großen Datensätzen wird. Außerdem bieten sie Funktionalitäten, wie die Verwaltung von Daten auf verteilten Systemen und das Scheduling der Prozesse. Außerdem entstehen unter Verwendung von WMS allgemein geringere Entwicklungskosten, da diese sonst für die Infrastruktur eines bestimmten Workflows eingesetzt werden müssten [1] [3].
Diese Arbeit befasst sich mit der Portierung eines spezifischen FORCE Workflows auf das WMS Snakemake. Dabei wird Snakemake im Hinblick auf Reproduzierbarkeit und Skalierbarkeit charakterisiert und die Vorteile und Herausforderungen einer solchen Portierung erfasst.

2. Zielsetzung

Ziel dieser Arbeit ist es, die Eigenschaften des auf Python basierenden WMS Snakemake zu charakterisieren. Dies soll auf Basis einer Portierung eines bestimmten praktischen, relevanten und nicht trivialen FORCE Workflow [1] auf Snakemake erfolgen. Dieser wurde bereits von Lehmann et al. zu Nextflow [3] und von Knapp zu Argo Workflow [6] portiert. Dabei soll besonderer Fokus auf dem Vergleich zwischen der Umsetzung auf verschiedenen WMS liegen, da sich diese im Detail stark unterscheiden können.

3. Forschungsstand

Der originale Workflow [2] wurde in FORCE programmiert, ein custom-made Framework zum Erstellen und Ausführen von EO (Earth Observation) Workflows auf eigenständigen Servern. Dieser konkrete EO Workflow analysiert die Langzeit-Vegetationsveränderungen der Mittelmeer-Insel Kreta. FORCE (Framework for Operational Radiometric Correction for Enviromental monitoring) [3] ist eine all-in-one (Bildverarbeitungs-) Engine die ARD (Analysis Ready Data) anhand von Satellitenbildern generiert. Dabei werden viele Schritte und Werkzeuge zu einem Workflow kombiniert [4] [5].
Bei der bereits beschriebenen Portierung auf Nextflow [3] wurde zwischen den Vorteilen und dem Aufwand einer solchen Portierung abgewogen und es wurde festgestellt, dass die Entwicklung in Nextflow EO Workflows skalierbarer und leichter reproduzierbar machen kann. Allerdings kam es bei der Ausführung auf verteilten Systemen zu Problemen und Inkompatibilitäten zwischen FORCE und dem WMS. Dadurch war es nötig, Workarounds zu konzeptionieren [3]. Im Bezug auf Snakemake und dem spezifischen Workflow ist noch nicht sicher, ob solche Inkompatibilitäten ebenfalls auftreten und dementsprechende Anpassungen nötig werden.
Snakemake ist ein WMS, dass reproduzierbare und skalierbare Datenanalysen in lesbarer, Python basierter DSL (Domain specific language) ermöglicht. Workflows können in Cluster, Grid und Cloud Umgebungen ausgeführt werden, ohne dass dazu eine Anpassung der Workflow Definition selber notwendig ist. Zusätzlich beinhalten Snakemake Workflows eine Auflistung aller erforderlicher Software, die dann automatisch in jeder Ausführungsumgebung vom WMS bereitgestellt wird [7] [8].

1. https://github.com/nf-core/rangeland
2. https://github.com/CRC-FONDA/FORCE2NXF-Rangeland#original-workflow
3. https://github.com/davidfrantz/force

In Snakemake teilen sich Workflows in einzelne Schritte auf, die mit sogenannten rules repräsentiert werden [1]. Jede Regel besteht aus Input-Dateien, Output-Dateien und einem ausführbaren Teil. Dieser führt entweder Shell Befehle, Python Code oder externe Skripts aus. Snakemake prüft für jeden Input eines Jobs, ob eine Regel existiert, die diesen Input generiert bzw. die Dateien bereits vorliegen [7][8]. Somit bestimmen die Output-Dateien und Input-Dateien der Regeln die Abhängigkeiten der einzelnen Schritte und es entsteht ein DAG (directed acyclic graph) aller Tasks. Der Scheduler entscheidet anhand des DAG dann, welche Jobs parallel abgearbeitet werden können und welche nacheinander ausgeführt werden müssen [2].

4. Umsetzung

Anhand der Erkenntnisse von Lehmann et al. [3] wird eine Portierung von FORCE auf Snakemake mit der von Nextflow (und anderen) verglichen. Dabei werden die Eigenschaften von Snakemake beobachtet und erfasst, wie diese das Vorgehen erschweren oder erleichtern, bzw. inwiefern sich die Umsetzung dieses Ports im Allgemeinen von der Umsetzung anderer Portierungen unterscheidet. Als Workflow soll der FORCE Workflow crc-fonda/rangeland bzw. nf-core/rangeland [2] portiert und auf einem Kubernetes Cluster verteilt ausgeführt werden.

Dabei wird zuerst ein Verständnis von FORCE bzw. dem Rangeland Workflow benötigt, sowie ein Erlernen von Snakemake und ein Verständnis der Portierung auf Nextflow (und anderen Portierungen), um adäquate Unterschiede in den Funktionsweisen erfassen zu können. Dann werden die einzelnen Schritte der Implementierung überprüft und Unterschiede und Probleme protokolliert. Zum Schluss soll ein Experience Report an Implementierungsschritten mit Schwierigkeiten und Vorteilen das WMS Snakemake charakterisieren und ein Fazit zum Aufwand, der Stärken sowie der Schwächen und potenziellen Workarounds aufgrund von Inkompatibilitäten zwischen FORCE und Snakemake vorliegen.

1. https://snakemake.readthedocs.io/en/stable/
2. https://github.com/nf-core/rangeland

5. Literatur

1. Badia R, Ayguade E, Labarta J. Workflows for Science: a Challenge when Facing the Convergence of HPC and Big Data. 15. März 2017. Supercomput Front Innov Int J. 4(1):27–47.

2. Schiefer C, Bux M, Brandt J, Messerschmidt C, Reinert K, Beule D, u. a. Portability of scientific workflows in NGS data analysis: a case study. ArXiv 2006.03104. Cornell University; 2020.

3. Lehmann F, Frantz D, Becker S, Leser U, Hostert P. FORCE on Nextflow: Scalable Analysis of Earth Observation Data on Commodity Clusters. 2021. In: Cong G, Ramanath M, Herausgeber. Proceedings of the CIKM 2021 Workshops. Gold Coast, Queensland, Australia.

4. Frantz D. FORCE [Internet]. 2023 [zitiert 3. Dezember 2023]. Verfügbar unter: https://github.com/davidfrantz/force

5. FORCE documentation — FORCE 3.0 documentation [Internet]. [zitiert 3. Dezember 2023]. Verfügbar unter: https://force-eo.readthedocs.io/en/latest/index.html

6. Knapp R. FORCE on Argo Workflow Studienprojekt Bericht. 12. August 2022.

7. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics. 1. Oktober 2012;28(19):2520–2.

8. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, u. a. Sustainable data analysis with Snakemake. 2021. F1000Research.