

研究発表

Dockerfile自動修正ツールの構築

辻 裕真

背景

Dockerなどのコンテナ技術はソフトウェアのインフラを構築する軽量仮想化技術として近年関心を集めている.

コンテナを構成するDockerfileのビルドは成功率が他の言語と比べて低く, 約34%ものDockerfileがビルドに失敗する.[1]

Dockerに関する研究はパフォーマンスに関わるものが多く, ビルドに関する研究はあまり行われていない.

[1]Jürgen Cito, Gerald Schermann, John Erik Wittern, Philipp Leitner, Sali Zumberi and Harald C. Gall. An Empirical Analysis of the Docker Container Ecosystem on GitHub. Proceedings of the IEEE/ACM 14th International Conference on Mining Software Repositories, pp. 323-333, 2017

背景

- Dockerfileのビルド失敗時に修正に要する時間
 - 124.5分（平均）， 44.2分（中央値）
 - C++(中央値：5分)やJava(中央値：12分)と比べるとかなり長い[2]



Dockerfileを自動修正できれば開発者の負担が軽減できる

[2]Yiwen Wu, Tao Wang, Yang Zhang and Huaimin Wang.

An Empirical Study of Build Failures in the Docker Context. Proceedings of the 17th International Conference on Mining Software Repositories, pp. 76–80, 2020

目的とアプローチ

- ・ 目的： Dockerfileの自動修正ツールを構築すること
- ・ アプローチ
 - ・ 過去のDockerfileの変更履歴を探索空間として、 その中から適切な修正を見つけ出し修正を適用する.

先行研究

- Dockerfileの自動修正ツールであるShipwrightを提案. [3]
- ビルド実行時のログの最後の数行をクラスタリングし, 修正できそうなエラーを目視で確認して人手で修正パターンを作成.
- ビルドに失敗したDockerfileのうち約20%を修正できた.

[3]Jordan Henkel, Denini Silva, Leopoldo Teixeira, Marcelo d'Amorim, Thomas Reps.
Shipwright: A Human-in-the-Loop System for Dockerfile Repair. Proceedings of the 43rd
International Conference on Software Engineering, 2021

先行研究の問題点

- ・ 先行研究で修正できたエラーは頻繁に発生するエラーであると考えられるが、全体の20%しか修正できていない.
- ・ 他の多くのエラーはクラスタリングできない多種多様なエラーであり、修正パターンを手動で作成するには限界がある.
- ・ 多数の変更履歴データから修正できる可能性の高い変更を見つけて適用することで、様々な種類のエラーに対応することができるのではないか.

アプローチ

探索空間

1. GitHub上のDockerfileを含むプロジェクトからCircleCIを使用しているプロジェクト特定する.
2. CircleCIでdocker buildを実行しているプロジェクトを特定する.
3. CircleCI APIを用いて過去のビルドデータを取得する.
4. ビルドが失敗から成功へと変わったタイミングのコミットを特定し, 修正前と修正後のDockerfileをASTで表現し差分を取得する.

アプローチ

修正の探索

1. ビルドに失敗するDockerfileをASTで表現する.
2. ASTをベクトル化し, 探索空間内の修正前のDockerfileのASTとのコサイン類似度を計算し, 順位付けを行う.
3. 順位の高い修正から順に適用し, ビルドが成功するか全ての修正でビルドが失敗するまで繰り返す.

現在行っていること

- ・ 自動修正についての勉強

取得した差分データから実際に変更を適用する方法等

- ・ データ収集

GitHub上のDockerfileを含むプロジェクトからCircleCIを使ったプロジェクトを抽出（完了）

その中からdocker buildを実行しているものを抽出