M-031

ファイルアクセス履歴からコンパイルを必要とする作業を推定する手法

A Method for Classifying Tasks that Require Compilation Using File Access History

野村 優文 † Masafumi Nomura
乃村 能成 † Yoshinari Nomura

1. はじめに

計算機のファイルアクセス履歴から計算機ユーザの作業内容を推定することで，利用すべきファイルの提示や作業に合わせたフォルダ構造の動的な変更を行うシステムが提案されている [1].

本稿では，ファイルアクセス履歴からユーザがコンパイルを必要とする作業をしていた期間を推定する手法について提案する．以降では，ファイルアクセス履歴の観点から見た作業の特徴について述べ，特徴を表現するファイル更新の間隔や繰返し頻度に関する閾値を決定した．また，これらを用いることにより，コンパイルの有無を正しく推定できた作業の数について報告する．

2. コンパイルを必要とする作業

2.1 特徴

コンパイルを必要とする作業は，ユーザによるソースコードの編集作業 (以下，**コーディング作業**) とコンパイラによるオブジェクトファイルの生成と結合作業 (以下，**ビルド作業**) からなる．また，コーディング作業とビルド作業は，交互に複数回繰り返されるという特徴を持つ．

コンパイルを必要とする作業をファイル更新の観点から見ると，コーディング作業には，比較的長い時間間隔で散発的なファイル更新が見られる一方で，ビルド作業には，短かい連続的な更新が見られる．両者のファイル更新の間隔 (以下，**ファイル更新密度**) の変化が作るパターンは，他の作業と異なるパターンを持つため，このパターンを抽出することで，コンパイルを必要とする作業をそれ以外の作業と区別できる可能性がある．

この考えに基づき，ファイルアクセス履歴中からコンパイルを必要とする作業で生じた履歴を推定する手法を示す．

2.2 推定手法

コンパイルを必要とする作業のファイルアクセス履歴の例を**図 1** に示す．たとえば，図 1 より，時刻 14:00:00 から 14:00:02 の間にファイル A，B，C を更新したことを示している．この区間を R_1 とする．続くファイル D〜H の更新を R_2 とする．R_1 と R_2 の間には，一定時間ファイルの更新がない．この時間が閾値を超えたところで 2 つの区間に分けている．この閾値を X とする．このようにして，X 秒間の間隔でファイル更新を

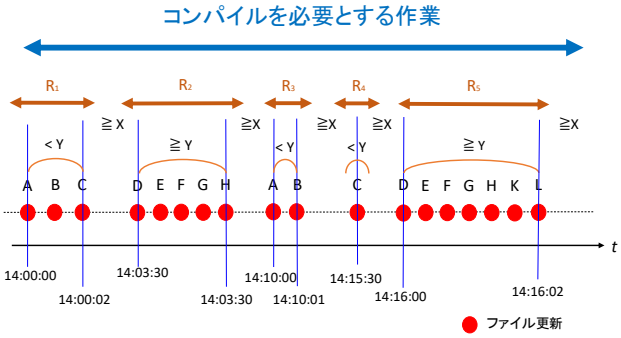

図 1 コンパイルを必要とする作業のファイルアクセス履歴の例

複数の区間に分割する．

分割されたこれらの区間には，そこに含まれるファイル更新の数が多い区間と少ない区間が存在する．そこで，ファイル更新の数が多い区間と少ない区間を閾値 (Y とする) を使用して分別する．分別した結果，ファイル更新の数が多い区間と少ない区間の繰り返しパターンが見られる場合，これら一連の作業をコンパイル作業として推定する．図 1 を用いて，以下にその詳細を示す．

(1) ファイル更新の間隔が X 秒以上ある点を境界として，ファイルアクセス履歴を R_1，R_2，...，R_m の区間に分割する．

(2) 各 $R_i(1 \leq i \leq m)$ に含まれるファイル更新の数を C_i としたとき，$C_i \geq Y$ であれば R_i をファイル更新密度が高い区間とし，他の区間は，ファイル更新密度が低い区間とする．

(3) 各 $R_i(1 \leq i \leq m)$ のうち，隣接する区間のファイル更新密度が共に高い，あるいは低い場合は，1 つの区間として結合する．結合した結果，新しい区間 S_1，S_2，...，$S_n(1 \leq n \leq m)$ を得る．たとえば図 1 の場合，R_3 と R_4 はファイル更新密度が低い区間が連続しているため，1 つにまとめる．

(4) 各 $S_i(1 \leq i \leq n)$ のうち，ファイル更新密度が高い区間と低い区間がそれぞれ 2 つ以上ある場合，$S_i(1 \leq i \leq n)$ 全体をコンパイルを必要とする作業として推定する．

† 岡山大学大学院自然科学研究科，Graduate School of Natural Science and Technology, Okayama University

表 1　コンパイルを必要とする作業の推定における閾値を変更したときの F 値

		ファイルアクセス履歴分割の閾値 (秒)									
		1	2	3	4	5	6	7	8	9	10
ファイル更新の回数の閾値 (回)	2	0.821	0.662	0.629	0.629	0.629	0.629	0.629	0.588	0.567	0.554
	3	**0.860**	0.775	0.748	0.724	0.707	0.698	0.685	0.667	0.648	0.620
	4	0.850	0.780	0.774	0.750	0.737	0.715	0.702	0.685	0.648	0.620
	5	0.850	0.805	0.784	0.760	0.742	0.720	0.711	0.694	0.657	0.619

表 2　コンパイルを必要とする作業の推定手法を使用したときの作業の推定結果

		推定結果がコンパイルを必要とするか		
		必要とする作業	必要としない作業	合計
正解データがコンパイルを必要とするか	必要とする作業	**184**	59	243
	必要としない作業	95	**337**	432
合計		279	396	**675**

3.　評価

3.1　閾値 X, Y の決定

2.2 節で述べた閾値 X と Y を決定する．まず，パラメータ調整用の教師データを用いて，X と Y を変化させながら，正解に対する適合率と再現率の調和平均である F 値が最高となる X，Y を決定した．

ファイルアクセス履歴分割の閾値 (X 秒) は 1〜10 秒の範囲で 1 秒ずつ変化させて評価した．下限を 1 秒としたのは，実験で使用したファイルアクセス履歴は，最小間隔が 1 秒として記録されるためである．

ファイル更新回数の閾値 (Y 回) は 2〜5 回の範囲で変化させて評価する．上限を 5 としたのは，ファイルアクセス履歴分割の閾値 (X 秒) の探索範囲上限である 10 秒間において，5 回以上ユーザが自らファイルを更新することは無いと考えるためである．

推定に利用するファイルアクセス履歴は，1 分以上継続して行われた作業の履歴を使用する．

閾値の決定には，2020 年 6 月 29 日〜2020 年 12 月 21 日の間で収集した 442,715 件のファイルアクセス履歴を用いた．X と Y を変化させたときの F 値を**表 1** に示す．表 1 より，ファイルアクセス履歴分割の閾値 (X 秒) を 1 秒とし，ファイル更新回数の閾値 (Y 回) を 3 回としたとき，0.860 となり，F 値は最大値だった．したがって，コンパイルを必要とする作業を推定する際は，これらの値を閾値として使用する．

3.2　評価結果

評価のため，著者が在籍する大学の研究室の学生 4 名が使用する 5 台の計算機を用いてファイルアクセス履歴を収集した．これらの計算機において，2020 年 10 月 1 日〜2020 年 12 月 31 日の間で収集した 408,510 件のファイルアクセス履歴を使用する．

文献 [1] で示された手法により，これらのファイルアクセス履歴をまず作業ごとに分割する．分割した作業ごとに含まれるファイルアクセス履歴に対し，提案手法により推定を行った．この推定には 3.1 節で決定した閾値を用いた．

この結果より，コンパイルを必要とする 1 分以上の作業が行われた数とコンパイルを必要としない 1 分以上の作業が行われた数を調べる．

推定結果を**表 2** に示し，以下に説明する．

(1) 243 個のコンパイルを必要とする 1 分以上の作業のうち，184 個の作業をコンパイルを必要とする作業であると正しく推定できた．

(2) 432 個のコンパイルを必要としない 1 分以上の作業のうち，337 個の作業をコンパイルを必要としない作業であると正しく推定できた．

上記の結果より，評価に使用した 675 個の作業のうち，521 個の作業のコンパイルの有無を正しく推定できた．

4.　おわりに

ファイルアクセス履歴からユーザがコンパイルを必要とする作業をしていた期間を推定する手法について提案した．提案手法は 408,510 件のファイルアクセス履歴より 675 個の作業を抽出し，このうち 521 個の作業におけるコンパイルの有無を正しく推定できた．

残された課題は，閾値 X を 1 秒未満の分解能で設定した際のコンパイルの有無の推定精度について調査することである．提案手法ではファイルアクセス履歴の最小間隔が 1 秒で記録される履歴を使用したとき，コンパイルの有無の推定に使用する閾値 X の最適な値は最小値である 1 秒だった．そこで，ファイルアクセス履歴を 1 秒未満の分解能で記録し，閾値 X を 1 秒未満の分解能で設定する．この際のコンパイルの有無の推定精度について調査する．

参考文献

[1] 西　良太，乃村能成：作業を代表するフォルダの推定と分類による仮想フォルダ生成システム，情報処理学会論文誌，Vol. 62, No. 2, pp. 527–537 (2021).