

A Dissertation for the Degree of Ph.D. in Engineering

Increasing Developer Productivity by
Improving Build Performance and
Automating Logging Code Injection

February 2021

Graduate School of Science and Technology

Keio University

Takafumi Kubota

主 論 文 要 旨

報告番号	㊦ 乙 第	号	氏 名	窪田 貴文
<p>主 論 文 題 名 : Increasing Developer Productivity by Improving Build Performance and Automating Logging Code Injection (ビルド高速化とログの自動挿入による開発者の生産性向上に関する研究)</p>				
<p>(内容の要旨)</p> <p>近年、システムソフトウェアの開発規模が爆発的に増大しているにつれ、より効率的な開発手法が求められてきている。例えば、大規模なソフトウェア開発では Continuous Integration (CI) という手法が事実上の標準となっている。CI では開発者が自身のコード変更を頻繁にメインリポジトリにマージし、そのたびに自動化されたビルドとテストが実行されている。</p> <p>しかしながら、この開発手法の効率を低下させる要因として、開発者がソフトウェアのロジックとは無関係な工程に時間が取られてしまっている点がある。本論文では次の 2 点に注目する。1) ビルド時間：近年、頻繁にビルドが発生する開発環境ではビルド時間は無視できないオーバーヘッドとなってきている。2) ログの挿入：挿入されているログの品質が高いほど、大規模なソフトウェアのデバッグ時間が短くなることが知られている。しかし、どこにどのようなログを挿入すればよいのかは開発者の専門知識と Engineering effort に大きく依存し、時間のかかる作業である。</p> <p>そこで、本論文では大規模な C++ プロジェクトのビルドパフォーマンス向上を目指したビルドシステム Cauldron とマルチスレッドな C プロジェクト向けのログの自動挿入ツール K9 を提案する。Cauldron はより洗練された Unity Build をサポートし、コンパイルされるファイル数に応じて動的にビルドの挙動を変更する。そして、従来の手法よりもビルドに要する時間を改善できることを示す。例えば、WebKit の Continuous Build においてビルド時間を 23% 削減できる。</p> <p>また、K9 ではマルチスレッドで共有されるデータ間で発生するデータ依存関係を考慮したログの自動挿入を行う。マルチスレッド環境では障害を発生させるスレッドとバグを実行したスレッドが異なる場合があり、スレッド間の依存関係を記録することがデバッグ時に重要となる。K9 はソースコード解析を用いてスレッド間のデータ依存関係を発生させる典型的なコード箇所を検出し、記録するログコードを自動挿入する。K9 よって挿入されたログは Linux kernel の 1 つの未発見のバグを含む、4 つの実在するバグに対してデバッグ時に有効な情報を提供できることを示す。</p> <p>本論文の貢献は次にまとめられる。ソフトウェア開発の効率を妨げる要因となるソフトウェアロジックとは無関係な工程であるビルドとログの挿入に対して有効な 2 つツールの提案・実装・評価を示している。ソフトウェア開発は複数の工程で成り立っており、開発サイクル全体の効率化を図ることが重要である。本論文では、ビルドとログの挿入の 2 つの異なる工程の効率化を模索しており、ソフトウェア開発のそれぞれの工程での効率化に役立てることができる。</p>				

Thesis Abstract

No. _____

Registration Number	<input checked="" type="checkbox"/> "KOU" <input type="checkbox"/> "OTSU" No. _____ *Office use only	Name	Takafumi Kubota
Thesis Title Increasing Developer Productivity by Improving Build Performance and Automating Logging Code Injection			
Thesis Summary <p>As software is growing in size and complexity, it is critical to develop software efficiently and reliably. For example, Continuous Integration (CI) has become a de-facto practice of the daily development in large software projects, in which builds and tests are automated, resulting in numerous developers' modifications are efficiently integrated in a mainline.</p> <p>However, as an obstacle for improving development efficiency, software engineers spend lots of time outside of the actual development of the software. This dissertation addresses two problems. 1) Build time: builds occur frequently during software development. As a result, the times spent on builds is a noticeable overhead. 2) Logging code insertion: the quality of log messages is critical to the efficiency in failure diagnosis. However, appropriately inserting the logging code is time-consuming because it depends on developers' expertise and engineering effort.</p> <p>To deal with these problems, I introduce two effective tools. For build times, I present a new build system, called Cauldron, which aims to improve the build performance for large C++ projects. Cauldron supports sophisticated unity builds and adaptive build behavior based on the number of files to be compiled. My experiments show that Cauldron outperforms existing approaches; for example, it reduces build times of WebKit by 23% in continuous builds.</p> <p>For logging code insertion, I introduce a new logging tool, called K9, which automatically inserts the logging code to trace inter-thread data dependencies caused by shared data among threads. In multi-threaded systems, the traceability of inter-thread data dependencies is essential in failure diagnosis because the thread actually causing the failure may be different from the thread executing the buggy code. In my experiments, I show that the log of K9 provides useful clues for debugging four bugs in the Linux kernel, including one unknown bug.</p> <p>The contribution of this dissertation is summarized as follows. As software development consists of multiple tasks, it is important to consider the various processes in the development cycle. This dissertation proposes two tools to improve the efficiency of two specific parts of software development: build time and logging code insertion. I describe in detail the design, implementation, and evaluations of the two tools.</p>			