

# Operating System Support for Safe and Efficient Auxiliary Execution

山下 恭平

**概要:** 本稿は USENIX, OSDI'22 に掲載されている論文「Operating System Support for Safe and Efficient Auxiliary Execution」<sup>[1]</sup> の内容についてまとめたものである。近年のアプリケーションは様々な補助タスクが実行されている。補助タスクとは、アプリケーションが自身のメンテナンス、自己管理を行うタスクのことである。これらのタスクは、アプリケーションのアドレス空間で実行することで、高い観測性と制御性を得るが、安全性と性能の問題が発生する。また、補助タスクを別のプロセスで実行すると、分離性は高いが、観測性と制御性が劣る。本稿では、この問題を解決するために、補助タスクに対するサポートとして、orbit と呼ばれる OS の抽象化機能を提案する。

## 1. はじめに

運用されているアプリケーションはその実行状況を調査し、最適化し、デバッグし、制御するために、頻繁にメンテナンスを行う必要がある。かつてはメンテナンスはアプリケーションの管理者が手動で行なっていたが、現在では多くのアプリケーションは、自身でメンテナンスを行うための補助タスクが行われている。例えば MySQL では、デッドロックを検出するとロールバックを行う機能が存在する。<sup>[2]</sup> そのため補助タスクはアプリケーションの信頼性や観測性に大きく影響する。既存の OS に搭載されているプロセスやスレッドといった抽象化機能はメインタスクの実行に適した設計がされており、補助タスクの実行には適していない。そのため、開発者は、分離は強いが観測と制御が非常に限定される（別プロセス）か、観測と制御は強いが分離はほとんどできない（スレッド）かのどちらかを選ばざるを得ない。この問題を解決するために、OS の補助タスクに対するサポートを提供する orbit 抽象化を提案する。

orbit タスクは、協力的な分離を提供する。同時に、状態同期機能によりメインタスクを観測することも可能である。orbit のプロトタイプは Linux kernel 5.4.91 に実装された。orbit の評価を行うために MySQL, Apache を含む 6 つの大規模アプリケーションから、7 つの補助タスクを抽出し、orbit に移行することに成功した。また、全てのケースでアプリケーションはフォールトから保護されることがわかった。分離のコストを測定した所、orbit バージョンアプリケーションでは、中央値で 3.3 % のオーバーヘッドが発生した。

本稿では 2 章でこの研究が行われる背景について述べ、3 章で orbit の詳細な説明を行う、4 章で orbit の性能評価を行い、最後の 5 章では全体のまとめる。

## 2. おわりに

### 参考文献

- [1] Yuzhuo Jing , Peng Huang . Operating System Support for Safe and Efficient Auxiliary Execution , 16th USENIX Symposium on Operating Systems Design and Implementation , page 633 - 648 (2022).
- [2] Oracle. MySQL's deadlock detection. <https://dev.mysql.com/doc/refman/8.0/en/innodb-deadlock-detection.html> (11/25/2022)
- [3] R. Wahbe, S. Lucco, T. E. Anderson, and S. L. Graham . : Efficient software-based fault isolation. In Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles, SOSP '93, page 203 - 216 , (1993).
- [4] A. Bittau, P. Marchenko, M. Handley, and B. Karp. Wedge: Splitting applications into reduced-privilege compartments. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI '08, page 309 - 322, (2008).
- [5] J. Litton, A. Vahldiek-Oberwagner, E. Elnikety, D. Garg, B. Bhattacharjee, and P. Druschel. Light-weight contexts: An OS abstraction for safety and performance. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI '16, page 49-64, (2016).
- [6] InnoDB deadlock detection is CPU intensive with many locks on a single row. <https://bugs.mysql.com/bug.php?id=49047> . (11/25/2022)