| | |
|---|---|
| Title | リアルタイム言語モデル |
| Sub Title | Real-time language model |
| Author | 川島, 英之(Kawashima, Hideyuki) |
| Publisher | 慶應義塾大学 |
| Publication year | 2022 |
| Jtitle | 学事振興資金研究成果実績報告書 (2021. ) |
| JaLC DOI | |
| Abstract | 言語モデルにはニューラルネットワークやベクトル空間モデルなど様々な実現方式がある．これをリアルタイム化するには，データ挿入の高速化が必要である．データはトランザクショナルにデータベースに書き込まれるため，データベースの応答を高速化することが重要である．主記憶データベースシステムでは永続性のためにログを永続化ストレージに書く必要があり，これが性能や応答時間に大きく影響する．<br><br>本研究では，応答時間を抑えるため，Siloプロトコルを対象にして応答高速化手法を提案した．Silo は高スループットを提供することが知られている．その並行性制御法は楽観法であるため，最適化を施さなければ高競合下でCicada 法などの近代的手法に劣るが，スロットリング最適化を利用すればCicada法に優ることも知られている．また，Silo<br>は近代的制御法の中で唯一プロダクションシステムに実装されている．従ってSilo<br>の応答時間改善は学術的にも実用的にも価値があると考えられる．<br><br>他方，Silo の応答時間はデフォルトで平均20 msと長い．この理由は，Silo はエポックに基づく一括永続化でスループット性能を高めようとするからであり，そのエポック更新周期がデフォルトで40 msだからである．そのナイーブな応答時間改善手法は，この更新周期を40 msよりも短くすることである．ところがこの方式では応答時間が改善しない．なぜならCPUコアは複数あるためロガースレッドも複数存在し，各ロガースレッドの進捗状況に大きな差が出る場合がある場合，応答時間が急激に劣化するからである．<br><br>そこで永続化が遅れているロガースレッドに対応するワーカスレッドの進行を抑制するエポック同期法なる手法を提案した．実証実験では，従来手法から提案手法にすることにより，<br>22%のスループット減少と引き換えに，97%減となる10.7 ms<br>の応答時間を達成した．すなわち，リアルタイム言語モデル基盤の構成技法を創出した．<br>There are various language models, such as neural networks and vector space models. To make this real-time, data insertion must be accelerated. Since data is written to the database transactionally, it is important to speed up the database response. Main memory database systems require logs to be written to persistent storage for persistence, which significantly affects performance and response time.<br><br>In order to reduce the response time, we proposed a response acceleration method for the Silo protocol, which is known to provide high throughput. Its concurrency control method is an optimistic method, which is inferior to modern methods such as the Cicada method under high competition without optimization, but it is known to be superior to the Cicada method if throttling optimization is used. Silo is also the only modern control method that has been implemented in a production system. Therefore, Silo's response time improvement is considered valuable both academically and practically.<br><br>On the other hand, Silo's response time is long, averaging 20 ms by default. The reason for this is that Silo attempts to improve throughput performance with epoch-based batch persistence, and its default epoch update cycle is 40 ms. The naïve response time improvement method is to make this update cycle shorter than 40 ms. However, this method does not improve response time. This is because there are multiple CPU cores and therefore multiple logger threads, and if there is a large difference in the progress of each logger thread, the response time will rapidly deteriorate.<br><br>Therefore, we proposed an epoch-synchronization method that suppresses the progress of worker threads corresponding to logger threads whose persistence is delayed. In a demonstration experiment, a response time of 10.7 ms, a 97% reduction, was achieved in exchange for a 22% reduction in throughput by switching from the conventional method to the proposed method. In other words, we have created a technique for constructing a real-time language model |

| | |
|---|---|
| | infrastructure. |
| Notes | |
| Genre | Research Paper |
| URL | https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=2021000004-20210040 |

2022 年 4 月 5 日

# 2021 年度　学事振興資金（共同研究）研究成果実績報告書

| 研究代表者 | 所属 | 環境情報学部 | 職名 | 准教授 | 補助額 | 1,200　千円 |
|---|---|---|---|---|---|---|
| | 氏名 | 川島 英之 | 氏名（英語） | Hideyuki Kawashima | | |

## 研究課題（日本語）

リアルタイム言語モデル

## 研究課題（英訳）

Real-Time Language Model

## 研究組織

| 氏　名 Name | 所属・学科・職名 Affiliation, department, and position |
|---|---|
| 川島英之（Hideyuki Kawashima） | 環境情報学部・環境情報学科・准教授 |
| 青山敦（Atsushi Aoyama） | 環境情報学部・環境情報学科・准教授 |
| 大磯一（Hajime Oiso） | 環境情報学部・環境情報学科・准教授 |

## １．研究成果実績の概要

言語モデルにはニューラルネットワークやベクトル空間モデルなど様々な実現方式がある．これをリアルタイム化するには，データ挿入の高速化が必要である．データはトランザクショナルにデータベースに書き込まれるため，データベースの応答を高速化することが重要である．主記憶データベースシステムでは永続性のためにログを永続化ストレージに書く必要があり，これが性能や応答時間に大きく影響する．

本研究では，応答時間を抑えるため，Silo プロトコルを対象にして応答高速化手法を提案した．Silo は高スループットを提供することが知られている．その並行性制御法は楽観法であるため，最適化を施さなければ高競合下で Cicada 法などの近代的手法に劣るが，スロットリング最適化を利用すれば Cicada 法に優ることも知られている．また，Silo は近代的制御法の中で唯一プロダクションシステムに実装されている．従って Silo の応答時間改善は学術的にも実用的にも価値があると考えられる．

他方，Silo の応答時間はデフォルトで平均 20 ms と長い．この理由は，Silo はエポックに基づく一括永続化でスループット性能を高めようとするからであり，そのエポック更新周期がデフォルトで 40 ms だからである．そのナイーブな応答時間改善手法は，この更新周期を 40 ms よりも短くすることである．ところがこの方式では応答時間が改善しない．なぜなら CPU コアは複数あるためロガースレッドも複数存在し，各ロガースレッドの進捗状況に大きな差が出る場合がある場合，応答時間が急激に劣化するからである．

そこで永続化が遅れているロガースレッドに対応するワーカスレッドの進行を抑制するエポック同期法なる手法を提案した．実証実験では，従来手法から提案手法にすることにより，22%のスループット減少と引き換えに，97%減となる 10.7 ms の応答時間を達成した．すなわち，リアルタイム言語モデル基盤の構成技法を創出した．

## ２．研究成果実績の概要（英訳）

There are various language models, such as neural networks and vector space models. To make this real-time, data insertion must be accelerated. Since data is written to the database transactionally, it is important to speed up the database response. Main memory database systems require logs to be written to persistent storage for persistence, which significantly affects performance and response time.

In order to reduce the response time, we proposed a response acceleration method for the Silo protocol, which is known to provide high throughput. Its concurrency control method is an optimistic method, which is inferior to modern methods such as the Cicada method under high competition without optimization, but it is known to be superior to the Cicada method if throttling optimization is used. Silo is also the only modern control method that has been implemented in a production system. Therefore, Silo's response time improvement is considered valuable both academically and practically.

On the other hand, Silo's response time is long, averaging 20 ms by default. The reason for this is that Silo attempts to improve throughput performance with epoch-based batch persistence, and its default epoch update cycle is 40 ms. The naïve response time improvement method is to make this update cycle shorter than 40 ms. However, this method does not improve response time. This is because there are multiple CPU cores and therefore multiple logger threads, and if there is a large difference in the progress of each logger thread, the response time will rapidly deteriorate.

Therefore, we proposed an epoch-synchronization method that suppresses the progress of worker threads corresponding to logger threads whose persistence is delayed. In a demonstration experiment, a response time of 10.7 ms, a 97% reduction, was achieved in exchange for a 22% reduction in throughput by switching from the conventional method to the proposed method. In other words, we have created a technique for constructing a real-time language model infrastructure.

## ３．本研究課題に関する発表

| 発表者氏名<br>（著者・講演者） | 発表課題名<br>（著書名・演題） | 発表学術誌名<br>（著書発行所・講演学会） | 学術誌発行年月<br>（著書発行年月・講演年月） |
|---|---|---|---|
| Masahiro Tanaka, Hideyuki Kawashima | Stable Low Latency Logging for Epoch-based In-memory Database | 2022 IEEE International Conference on Big Data and Smart Computing (BigComp) | Jan. 2022 |

| | | | |
|---|---|---|---|
| Takayuki Hoshino, Suguru Knoga, Masashi Tsubaki, Atsushi Aoyama. | Comparing subject-to-subject transfer learning methods in surface electromyogram-based motion recognition with shallow and deep classifiers. | Neurocomputing. | In press. |