

電子商取引サイトにおける応答待ち時間の短縮を目的とした  
データベース参照のスケジューリング方式

Database Query Scheduling

to Improve the Response Time in an Electronic Commerce Site

赤坂 謙二郎<sup>†</sup>                      布目 淳<sup>‡</sup>                      平田 博章<sup>‡</sup>                      柴山 潔<sup>‡</sup>  
Kenjiro Akasaka              Atsushi Nunome              Hiroaki Hirata              Kiyoshi Shibayama

## 1. まえがき

ウェブショップなどの電子商取引サイト (Electronic Commerce Site: 以下 EC サイト) は、ウェブサーバとそのウェブサーバ上で動作するウェブショップ構築システム、およびデータベースサーバを主なコンポーネントとして構成される。クライアント (顧客) からの HTTP アクセスに対して、EC サイト内では、必要な情報をデータベースサーバに問い合わせ取得し、これをもとに動的にウェブページを生成する。EC サイトを効率良く機能させるためには、ネットワーク回線速度とウェブサーバ、データベースサーバの性能をハードウェアとソフトウェアの両面でバランスさせなければならないが、一般に、これは容易ではない。特に、人気商品の発売開始時やタイムセールなど、平常時とは比較にならないほど多くのアクセスが集中すると、データベースサーバの応答待ちで EC サイトが一時的に機能しなくなる状況も生じる。

本稿では、ウェブショップ構築システムとデータベース管理システムにそれぞれ Zen Cart[1]と MySQL[2][3]を用いた EC サイトを対象として、データベースサーバに対するウェブサーバからのクエリをスケジューリングすることにより、特にアクセスが集中する状況下において EC サイトの応答時間 (データベース処理時間を含むウェブページの生成時間) を短縮する方式を提案する。

## 2. EC サイトにおけるデータベースアクセス

Zen Cart を用いた EC サイトでは、商品に関する情報や在庫数、顧客情報などはすべて MySQL で管理するデータベースに統合して記憶される。

MySQL では、データベースからの読み出しのみを行うクエリ (以下、読み出しクエリ) については、複数のクエリを並列 (並行) に処理可能である。しかし、データの更新や追加、削除など、データベースへの書き込みを伴うクエリ (以下、書き込みクエリ) に対しては、原則としてテーブル単位で排他ロックをかけて処理を行う。書き込みクエリを受け付けると、データベースサーバはそのテーブルを参照するクエリの処理がすべて完了するのを待って、書き込み処理を開始する。また、その書き込みクエリの処理が完了するまで、同じテーブルを参照する他のクエリの処理を待たせる。したがって、このような待機が多発すれば EC サイトの応答時間が増大する。

EC サイトのクライアントが商品を買物カゴに入れたり、

実際に購入手続きを行うと、買物カゴの内容や商品の在庫数の値を更新するための書き込みクエリが発生する。これは上記の待機を生じさせる要因となり得るが、この書き込みは商取引上の本質的な処理であり、また、多くの注文が短時間に集中しなければ、性能上の問題とはならない。クライアントからのアクセスの多くは、商品や商品リストのページを要求するもの (商品を購入する場合でも、まずはその商品ページを表示しなければならない) であり、本来、ウェブサーバはデータベースから必要なデータを読み出すだけでよい。しかし、実際には、販促等の目的で商品ページごとにその閲覧回数をカウントすることが多く、この場合、ウェブサーバは、クライアントからのアクセスのたびにデータベース内の閲覧回数の値を更新するクエリを送信する。データベースに対するこのような書き込みクエリが他の読み出しクエリと混在して送信されるので、単に商品ページを表示するためだけのアクセスであっても、アクセス数が増えるとクエリ処理の待機が多発し、EC サイトの応答時間が増大する。

従来より、データベース処理の性能改善には、(1) テーブル構造や SQL プログラミング上の改善、(2) データベースサーバのメモリ使用量やスレッド数などのパラメータの最適化、などで対処されてきた。本稿で着目している待機の問題は、データベースシステム内部の処理方式として改善されるべき課題ではなく、EC サイトでの使用法との関連で発生するため、上記の従来のアプローチでは解決できない。また、ロック操作をテーブル単位ではなく行単位で行うことにすれば、待機の発生頻度を低減することが可能であるが、Zen Cart に限らず、osCommerce[4] や CubeCart[5]なども含めて、主にトランザクション管理の観点からそのような選択肢の採用が早い時期から予定されていたが、実際には実現に至っていない。そこで、本稿では、従来とは全く異なるアプローチによって EC サイトの性能改善を図る。商取引上で不都合を生じない範囲で、ウェブサーバが発する書き込みクエリと読み出しクエリの順序を入れ替え、複数の読み出しクエリを連続してデータベースサーバに送信することにより、データベースへの書き込みに伴う待機の発生頻度を低減する。

## 3. クエリのスケジューリング

### 3.1 スケジューリング方式の概要

本方式では、図1に示すように、ウェブサーバとデータベースサーバとの間にスケジューリングサーバを設置する。スケジューリングサーバは、ウェブサーバに対しては自らがデータベースサーバであるかのように振舞い、また、データベースサーバに対しては自らがウェブサーバであるかのように振舞う。ウェブサーバから受信したクエリの順序

<sup>†</sup> 京都工芸繊維大学大学院 工芸科学研究科 情報工学専攻

<sup>‡</sup> 京都工芸繊維大学大学院 工芸科学研究科 情報工学部門

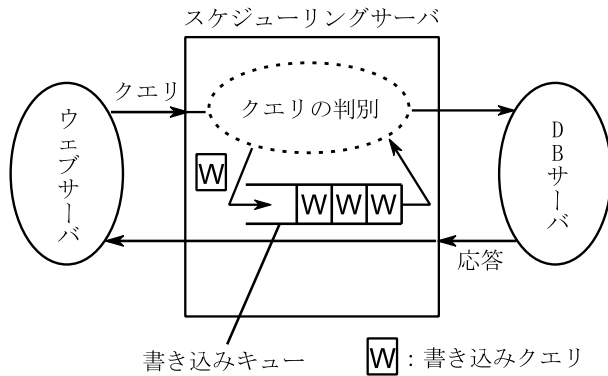


図1 スケジューリングサーバの設置

を入れ替えてデータベースサーバへ転送し、また、データベースサーバからの応答メッセージをそのままウェブサーバへ転送する。

スケジューリングサーバ内には、ウェブサーバから受信した書き込みクエリを一時的に保存するためのキュー（以下書き込みキュー）を設ける。商品ページ等の閲覧回数を更新するための書き込みクエリを書き込みキューに一時的に格納し、これに後続する読み出しクエリ等を先にデータベースサーバへ転送する。書き込みキューにクエリが保存されている間は、商品ページの実際の閲覧回数とデータベースに記憶されている閲覧回数との間に不整合が生じるが、このような情報は、商品の在庫数等とは異なり、リアルタイムに更新される必要性はない。したがって、このような書き込みクエリに対する実際のデータベース処理を遅らせても、商取引上で実質的な問題は生じない。

以上のように、ウェブサーバからの書き込みクエリをスケジューリングサーバによって一時的に保存することで、読み出しクエリを連続してデータベースサーバへ転送することができ、その結果、書き込みクエリが読み出しクエリの中に散発的に混在することによって引き起こされるクエリ処理の待機の影響を抑えることができる。

### 3.2 スケジューリングサーバの処理内容

スケジューリングサーバは、図2に示すように、ウェブサーバからクエリを受信すると、まずそれが読み出しクエリであるか書き込みクエリであるかを判別する。受信したクエリが読み出しクエリの場合はそれをすぐにデータベースサーバへ転送し、書き込みクエリの場合はその書き込みクエリが書き込みキューへのキューイング対象であるかどうかを判別する。そして、キューイング対象の書き込みクエリでなければそのままデータベースサーバへ転送し、キューイング対象の書き込みクエリであればスケジューリングサーバ内の書き込みキューに格納する。ただし、ウェブサーバは、書き込みクエリに対する応答を受信するまでページ生成処理を中断してしまうので、スケジューリングサーバは、キューに保存する書き込みクエリに対してデータベースサーバが返すであろう応答メッセージを、あらかじめこの時点でウェブサーバに送信する。

受信した書き込みクエリがキューイングの対象かどうかを判別するには、本来、SQL文を解析する必要がある。しかし、スケジューリングサーバはウェブサーバとデータベースサーバとの間に位置し、スケジューリングサーバで扱

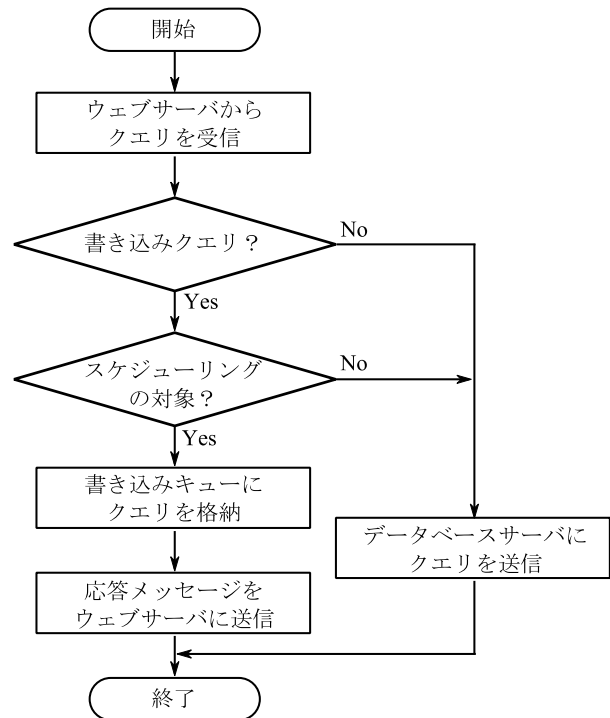


図2 スケジューリングサーバの処理の流れ

うクエリはウェブサーバ上で動作するウェブショップ構築システムから送信されるものに限定されるため、書き込みクエリに対して複雑な解析を行う必要はない。つまり、スケジューリングサーバは、簡単なパターンマッチングによって、受信した書き込みクエリがキューイングの対象かどうかを判別することができる。

書き込みキューに保存したクエリは、以下に挙げるいずれかの時点でデータベースサーバへ送信する。このとき、書き込みキュー内のすべてのクエリを連続して送信する。

- 1) 書き込みキューに最初のクエリを格納してから、あらかじめ設定した時間（最大保存時間）が経過したとき。
- 2) 書き込みキュー内のクエリの数が、あらかじめ設定した最大数（最大保存件数）に達したとき。
- 3) ウェブサーバからクエリを最後に受信してから、あらかじめ設定した時間（最大空白時間）が経過したとき。
- 4) 転送したすべてのクエリに対する応答メッセージをデータベースサーバから受信したとき。
- 5) キューイングの対象でない書き込みクエリをデータベースサーバに転送した直後。

## 4. 性能評価

### 4.1 評価環境

Zen CartとMySQLを用いて実験用のECサイトを構築し、このECサイトに対して、本稿で提案するクエリのスケジューリング方式を実際に適用して、性能評価を行った。ある実在のネットショップにおける新着商品発売時の状況を想定して、図3に示す環境で応答時間を測定した。図3中の各クライアントはそれぞれ以下のように動作する。

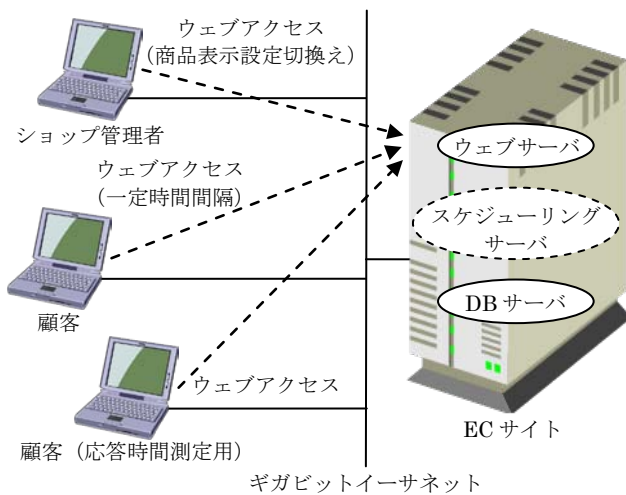


図3 測定環境

- ショップ管理者クライアント：新着商品の表示設定の切替をウェブアクセスによって繰り返す。
- 顧客クライアント：ECサイトに対して、新着商品一覧のページを一定時間間隔で要求する。1回のアクセスで数千のクエリを発生させる。
- 顧客クライアント（応答時間測定用）：新着商品一覧のページを要求し、ウェブサーバ上で測定したその応答時間を収集する。したがって、ここでの応答時間とはウェブサーバによるページ生成時間を指し、クライアントとECサイトとの間のネットワーク通信時間は含まない。複数回測定を行い、以下ではその平均値を用いて議論する。

#### 4.2 スケジューリングサーバのパラメータ

スケジューリングサーバについて、最大保存時間、最大保存件数、最大空白時間の各パラメータを変化させたときのECサイトの平均応答時間を、それぞれ図4、図5、図6に示す。図中のアクセスレートは、顧客クライアントからのHTTPアクセスが1秒間に何回行われるかを示す。また、あるパラメータに関して測定を行う際には、他の2種のパラメータには十分に大きな値を設定し、それらのパラメータが応答時間に影響しないようにした。なお、図6においてグラフの一部が欠落しているが、これは、アクセスレートを0.41回/秒以上としたときにはウェブサーバから発行されるクエリの量が多くなり、0.3秒以上の空白時間が発生しなくなったためである。

図4～6のグラフより、各パラメータの値を大きくするにともなってECサイトの平均応答時間を短縮できるが、それぞれある値を超えて大きく設定しても、平均応答時間はそれ以上短くならないことがわかる。

#### 4.3 読み出しクエリの連続転送数

4.2節の結果から、書き込みクエリに着目して設けた送信条件に関して、各パラメータの最適な組み合わせを決定することは難しい。そこで、読み出しクエリを連続してデータベースサーバへ転送することによってクエリ処理の待機の発生を抑えるという観点から、3.2節の3つの条件1)～3)の代わりに、以下に示す送信条件を新たに設ける。

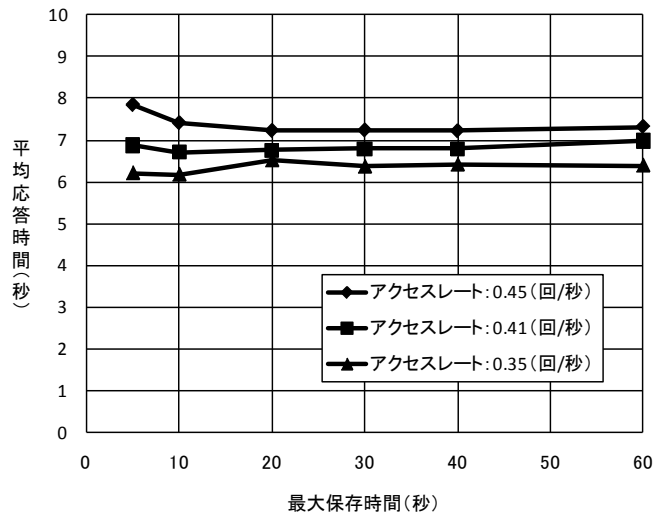


図4 最大保存時間による平均応答時間の変化

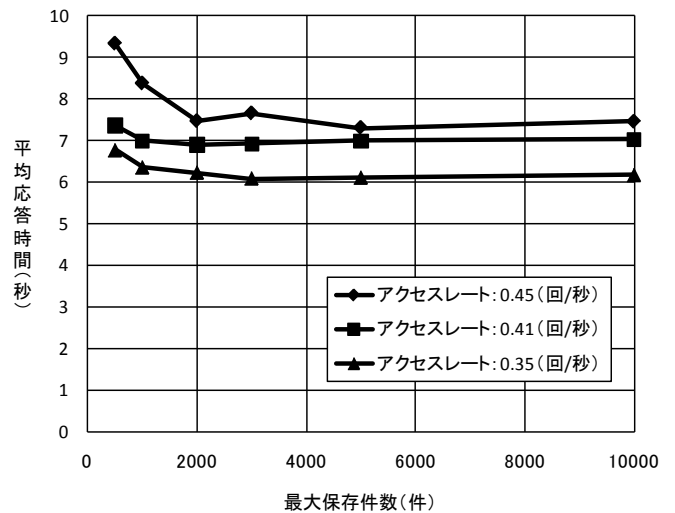


図5 最大保存件数による平均応答時間の変化

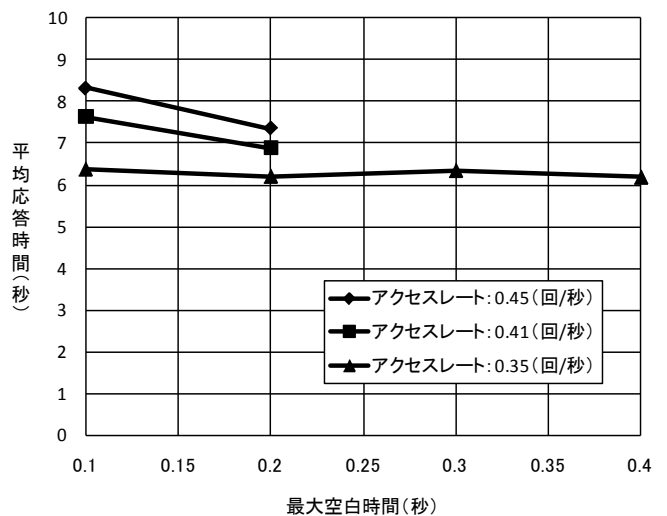


図6 最大空白時間による平均応答時間の変化

- 6) ウェブサーバからの読み出しクエリを連続して転送した数が、あらかじめ設定した数に達したとき。

この送信条件と4)、5)を用いてスケジューリングを行い、読み出しクエリの連続転送数を変化させたときのECサイトの平均応答時間を測定した。その結果を図7に示す。4.2節と同様に、ECサイトへのアクセスレートを3段階に設定して測定を行った。

図7より、読み出しクエリの連続転送数をパラメータとして用いることで、最大保存時間や最大保存件数、最大空白時間を用いるよりもスケジューリング効果を上手く制御できることがわかる。また、このパラメータを4,000以上にしても、平均応答時間は短くならないことから、ウェブサーバからの読み出しクエリを4,000件連続して転送した時点で、書き込みキュー内のクエリをすべてデータベースサーバへ送信するようにするのが、書き込みクエリの処理の遅延とECサイトの性能とのトレードオフの観点から望ましいことがわかる。

#### 4.4 クエリスケジューリングの効果

最後に、実験用のECサイトに対してスケジューリングサーバを適用する場合としない場合とでECサイトの応答時間を比較する。

顧客クライアントによるECサイトへのアクセスレートを変化させ、スケジューリングサーバを用いる場合と用いない場合のECサイトの平均応答時間を測定した。その結果を図8に示す。スケジューリングサーバを用いる場合の読み出しクエリの連続転送数のパラメータは、4.3節の調査結果より4,000に設定した。

図8に示すように、スケジューリングサーバを用いない場合、アクセスレートが0.4回/秒を超えたあたりから平均応答時間が急増した。これに対して、スケジューリングサーバを用いる場合は、アクセスレートの上昇に応じて平均応答時間は徐々に増大するものの、急激な応答時間の増大は抑えることができています。これより、ECサイトにアクセスが集中する状況では、クエリのスケジューリングが応答時間の短縮に有効であることがわかる。

一方、ECサイトへのアクセスレートが0.25回/秒より小さいときは、クエリのスケジューリングを行わない場合よりも平均応答時間が長い。しかし、その差はわずかであり、実際のウェブページ表示時間においては無視できる程度である。アクセスレートが小さいと、書き込みクエリによる他のクエリの待機があまり発生しないので、クエリをスケジューリングする効果が小さく、逆に、ECサイト内部でのプロセス間通信によるオーバーヘッドが顕在化したものと考えられる。ただし、このことは、スケジューリングのオーバーヘッドが十分に小さいことを示している。

#### 5. むすび

ECサイトにおいては、データベースに対する書き込みと読み出しの順序を入れ替えても、商取引上で不都合を生じない場合がある点に着目し、クライアントに対するECサイトの応答時間を短縮するために、ウェブサーバからのクエリをスケジューリングしてデータベースサーバへ送信する方式を提案した。これにより、ECサイトに多くのアクセスが集中する状況において、ECサイトの応答時間が極端に増大することを回避できることを確認した。

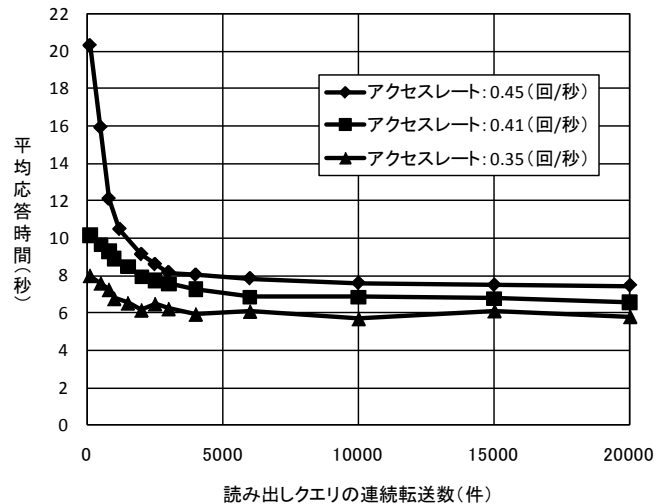


図7 読み出しクエリの連続転送数による平均応答時間の変化

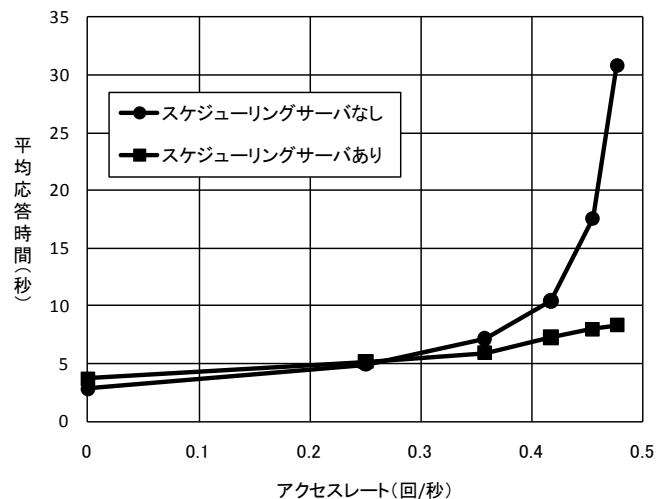


図8 クエリスケジューリングの効果

今回は、あるネットショップサイトにおける特定の状況を想定して性能評価を行ったが、今後は、他の環境や状況に対して広く評価を行う予定である。

#### 謝辞

本研究の一部は日本学術振興会科学研究費補助金（基盤研究 (C) 21500053 および同 22500046）による。

#### 参考文献

- [1] Zen Ventures, LLC, "ecommerce shopping cart software by Zen Cart ecommerce solution", <http://www.zen-cart.com/>
- [2] Sasha Pachev, 伊藤 直也, 田中 慎司, 吉川 英興, 菅野 良二, "詳解MySQL", オライリー・ジャパン (2007)
- [3] Sun Microsystems, "MySQL: The world's most popular open source database", <http://www.mysql.com/>
- [4] Harald Ponce de Leon, "Open Source E-Commerce Solutions", <http://www.oscommerce.com/>
- [5] Devellion Limited, "E-Commerce Software: CubeCart", <http://www.cubecart.com/>