

非定型ログ向けデータベースにおけるデータロード方式

森 郁海[†] 中村 隆顕[†] 加藤 守[†] 郡 光則[†]

三菱電機株式会社 情報技術総合研究所[†]

1. はじめに

近年の情報セキュリティ、省エネルギー意識の向上に伴い、センサデータやログデータなどの時系列・追記型データの活用が高まっている。我々は、時系列・追記型データを高速に処理するための高性能並列情報検索技術[1]を基盤として、非定型ログ向けデータベース（以下、「LDB」と表記）を開発した[2]。LDBは、効率的データ配置と高速・高圧縮率データ圧縮によるストレージアクセス技術、データ駆動型の並列処理技術などにより、多様な形式を持つログや非構造データなどの高速処理を実現する。また、LDBは1時間～1日に1回のようにバッチ的にデータを書き込む方法を採用する事で、効率よくデータロードを実施していた。一方で、ログ発生から即時利用するニーズが高まっており、ログデータの発生から利用可能になるまでの反映時間（以降、「反映時間」と表記）を短縮することが要求されている。これに対応するため、バッチの実行間隔を短縮する必要があるが、少量のデータ書き込みが頻発し、全体のスループットが低下するという課題があった。そこで、我々は、反映時間を削減しつつ、全体のスループットの低下を抑止するデータロード方式を開発した。本稿では、本データロード方式の概要と評価について説明する。また、性能評価を実施し、提案方式の効果を確認する。

2. 従来方式と課題

図1に従来の定期バッチ処理によるデータロードモデルの動作図を示す。まず、ログの発生源（以下、「クライアント」と表記）においてログが発生すると、ログ管理サーバ（以下、「サーバ」と表記）へログを送信する。サーバは、ログを受信すると、それを蓄積しておき、予め計画した定期バッチ処理によって、指定したLogSet（LDBにおけるログを管理するためのデータ蓄積単位）[2]へのロード命令を実行する。この命令によりログがLogSetに反映されると、ユーザからの利用が可能となる。また、各LogSetに対して、書き込み処理と競合しないよう予め計画した期間毎にバックアップ処理を実施する必要がある。

図2に従来方式のログの発生からログがデータ

ベースへ書き込まれるまでの処理時系列を示す。反映時間 $T_{reflection}$ を(式1)のように定義する。

$$T_{reflection} = T_{pre} + T_{send} + T_{delay} \quad (式1)$$

T_{pre} はログ送信の準備時間、 T_{send} はログ送信時のネットワーク遅延時間である。 T_{delay} は、サーバにログが到着してからユーザが利用可能となる時間である。 T_{delay} は、(式2)の関係が成り立つ。

$$T_{delay} = T_{wait} + T_{load} \quad (式2)$$

T_{delay} の支配的なパラメータは T_{wait} ($0 < T_{wait} < T_{exe}$) であり、バッチ処理の実行間隔 T_{exe} に依存する。即ち、反映時間を短縮するには、 T_{exe} を短縮する必要があるが、少量データのロードが頻発し全体の書き込みスループットが低下するという課題がある。

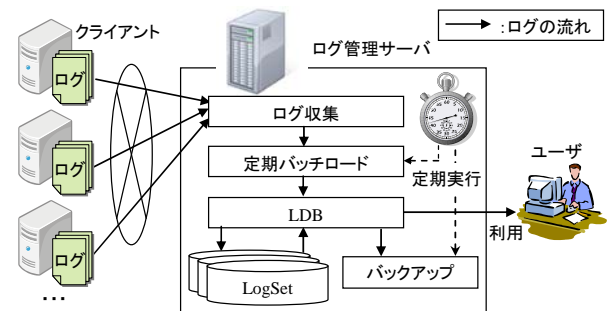


図1 従来方式(定期バッチロード)

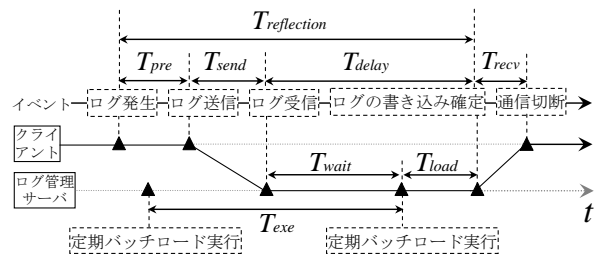


図2 従来方式の処理時系列

3. 提案方式

図3に本方式のデータロードモデルの動作図を示す。まず、クライアントにおいてログが発生すると、データロードクライアントAPIがサーバへログを送信する。サーバはログを受信すると、まずレコード生成処理を行う。次に、以下(1)または(2)の条件が成立すると、ログの書き込み処理が実行され、LogSetに反映される。

A Data Loading Scheme for the Atypical Log Database.

[†]Ikumi MORI Takaaki NAKAMURA Mamoru KATO Mitsunori KORI

Information Technology R&D Center, Mitsubishi Electric Corporation

- (1) *Tallowance* が経過した
 - (2) クライアントが書き込み確定命令を実行した
- Tallowance* は、許容する遅延時間として予めユーザが設定しておくものである。この処理により、*Tallowance* が経過するか書き込み確定命令を実行するまでの間、ログを蓄積しておくことで、LogSet に対する書き込み処理回数を抑制する。

図 4 に示す処理時系列を見ると、(式 2) が次のように変更されるのが分かる。

$$T_{delay} = Tallowance + Tload \quad (式 2')$$

T_{delay} の支配的なパラメータは $Tallowance (\geq 0)$ となる。但し、*Tallowance* は LogSet の排他制御 (図 5) のため、実際の処理完了時間を保証することはできない。

本方式でのバックアップ処理は、以下の手順で行う。この処理を予め計画した期間毎に実施する。

- (1) 範囲¹ [2] の追加
- (2) 過去の範囲のバックアップ
- (3) 過去の範囲の削除

また、LogSet に対する排他制御 (図 5) を実施することで、書き込み処理との競合を防ぐ。

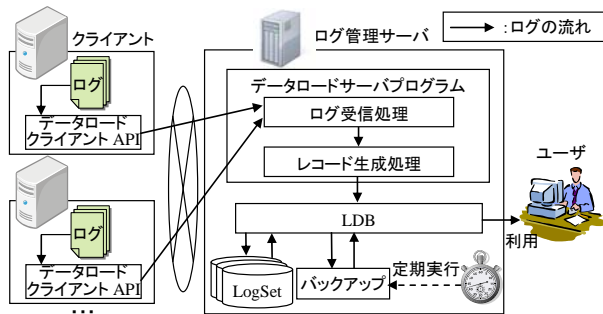


図 3 提案方式

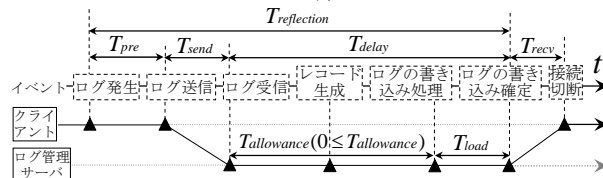


図 4 提案方式の処理時系列

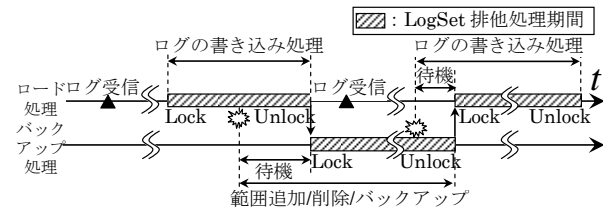


図 5 LogSet に対する排他制御

4. 評価と考察

本方式に処理ボトルネックが存在するかを確認するために、LDB 単独のロード性能との比較を実施

する。ロード性能の基準として、本方式のロード速度 (*Sload*) を (式 3) のように定義した。

$$Sload [MB/sec] = Totlog / (Treflection + Trecv) \quad (式 3)$$

Totlog は、LogSet に書き込んだログの総量である。

複数クライアントから常時ログをロードする状態を模擬するため、8 クライアントそれぞれにおいて、4,396.7 [MB] のログ (6,075,160 レコード、平均ログ長 723.7 [Byte]) を 10 [MB] ずつ分割して送信し、最後に書き込み確定命令実行するようにした ($Tallowance = \infty$ とした)。従って、 $Totlog = 4,396.7 [MB] \times 8$ となる。また、本方式の妥当性を評価するため、ネットワークがボトルネックにならないように、サーバとクライアントを同一ハードウェアに配置した (図 6)。8 多重接続時のループバックアドレスへのデータ転送速度は、1 接続あたり約 44.2 [MB/sec] である。表 1 に評価ハードウェア構成を示す。本評価では、バックアップ処理は実施しない。

測定の結果、 $Sload = 100.1 [MB/sec]$ であり、LDB 単独の LogSet への書き込み速度 157.0 [MB/sec] の約 64% の性能であった。

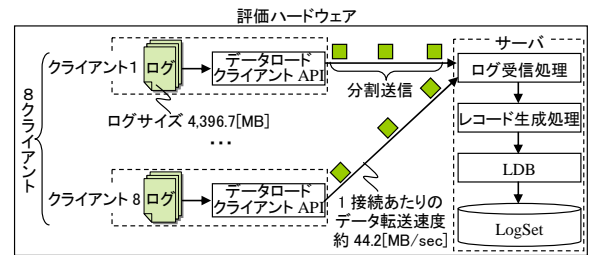


図 6 評価システム構成

表 1 評価ハードウェア構成

プロセッサ	3.33GHz Xeon (4 コア) X5470×2
メモリ	42.0GB
ディスク	300GB 15000 回転 U320 SCSI ×8 台
OS	Windows Server 2008 R2 Enterprise Edition

5. まとめ

LDB におけるデータロード方式の概要と評価について説明した。本方式により、ログ発生から実時間的にログを利用することが可能となった。

参考文献

- [1] 郡光則, 中村隆頭, 山岸義徳: 高性能並列情報検索技術, 三菱電機技報, 83, No.12 (2009).
- [2] 中村隆頭, 山岸義徳, 竹内丈志ほか: 大規模ログデータベースの実現, 第 68 回情報処理学会全国大会, Vol.68, No.3, pp.29-30 (2006).

¹⁾ 1 つの LogSet を時系列に従って複数の領域に区切ったもの。この範囲単位で検索、削除やバックアップ/リストアが可能。