D-022

センサデータのための分散サンプリングストレージとそのサンプルサイズ制御手法

Distributed Sampling Storage for Sensor Data and Its Sample Size Control Method

佐藤 浩史 †‡ Hiroshi SATO　井上 武 † Takeru INOUE　山崎 敬広 † Takahiro YAMAZAKI　片山 陽平 † Yohei KATAYAMA　清水 敬司 † Takashi SHIMIZU　小柳 恵一 ‡ Keiichi KOYANAGI

1　はじめに

今後 10 年で「モノのインターネット」が発達し，実空間に偏在する各種のセンサがネットワークに接続し，人や物，そして環境の情報を絶え間なく発信し続けることになると言われている [1]．この巨大なセンサデータ群を各種のアプリケーションに利用するためには，大量に流れ続けるデータを高速に保存・蓄積する分散ストレージ技術が重要性を増していくと予想されており [2]，研究が盛んに行われている．

一般的にデータベースにおいてはデータの永続性が重要視され，そのために冗長化や分散トランザクション管理などの多大なコストがかけられる．これは分散ストレージでも同じであり，大規模化を難しくする要因となっている．一方，センサデータのような膨大かつ連続的なデータに対するとき，多くのアプリケーションでは，個々のレコードよりもデータ全体の特徴量や統計的な傾向を捉えることが重要視される．例えば，平均値や分散，異なり数，変化率などである．また，それらの値も確率解や近似解を使うことが多い [3] ため，サンプリング処理を施すことが一般的であり，レコード個々の永続性はさほど重要ではないと考えられる．

そこで我々は，センサデータの特性を踏まえ，永続性を緩和したシンプルな分散ストレージ技術の研究を進めている [4]．このストレージはデータの統計的利用を前提としており，サンプリングの特性を利用して規模拡張性ならびに可用性・障害耐性を高めたアーキテクチャが特徴である．本稿ではその基本的なアーキテクチャと，サンプルサイズの制御手法について述べる．

2　分散サンプリングストレージ

2.1　要件

増え続けるデータおよびアクセスへ対応するために，規模拡張性および可用性は必須である．一方，既に述べたように，個々のレコードごとの永続性は要求されない．但し，指定された特徴量を正しく算出するのに十分なデータを，偏りなく保持していなければならない．また，特徴量の種類および要求される精度は利用するアプリケーションにより様々であるため，一般のデータストリーム処理で見られるようにデータの要約のみを保持する方法は好ましくない．データそのものを保持する必要がある．しかしその一方，常に全データを返すことは，ストレージおよびネットワークにかかる負荷を考慮すると，得策ではない．要求された精度とその処理によりかかる負荷が，トレードオフの関係になることが望ましい．

2.2　アプローチ

複数のデータベース（データサーバ）にデータを水平分割する分散構成を取る．その際，各データサーバ間には固定された上下関係を持たせず，互いの状態も監視せず，それぞれ独立にクライアントからのアクセスを受け付ける．そして，クライアントがデータを書き込む際のデータサーバの選択を，ランダムに行うものとする．これにより，書き込まれた各レコードはランダムにデータサーバに割り振られ，結果として各データサーバには全データに対するランダムサンプルが独立に蓄えられる．つまり，ストレージ自身にランダムサンプリング機能が内包されることになる．このようなストレージを分散サンプリングストレージと呼ぶことにする．

2.3　基本アーキテクチャ

本ストレージは，複数のデータサーバと管理サーバから成る (図 1)．管理サーバは，データサーバの死活監視や負荷分散に加えて，クライアントからのアクセスに対してデータサーバを指定する役割を担う．

レコードを書き込む際は，クライアントは管理サーバへその旨を伝える．管理サーバは正常に稼動しているデータサーバからレコード数分を重複を許してランダムに選び，そのアドレスをクライアントに伝え，クライアントは自らデータサーバへ書き込む．なお，管理サーバの負荷を減らすため，クライアントが管理サーバからデータサーバのリストを定期的に同期およびキャッシュし，自身で（ランダムに）データサーバ選択を行うこともできる．

統計量を読み出す際は，クライアントはクエリとして，統計量の種類と要求する精度，そして検索条件を管理サーバに伝える．管理サーバは要求精度から必要なサンプルサイズを見積り，負荷分散を考慮した上でデータサーバを必要な数だけ選択し，うち 1 台を代表サーバに指定し，代表サーバにクエリと他のデータサーバのアドレスリストを渡す．代表サーバは検索条件を各データサーバに投げ，適合するレコードを集め，統計量の計算を行い，結果をクライアントに返す[*1]．4 節で，サンプルサイズの見積りとデータサーバ数の決定について詳しく説明する．

3　評価

本節では，分散サンプリングストレージのシステムとしての特徴を定性的に評価する．ポイントとなるのは，各データサーバが独立に動作し，かつ各々が統計的に等質なデータ集合を蓄積していることである．

3.1　精度と負荷

各データサーバにはランダムサンプルが蓄積されているので，読み出し時に対象データサーバ数を調整することで，改めてサンプリング処理を行わなくても計算対象サンプルのサイズを制

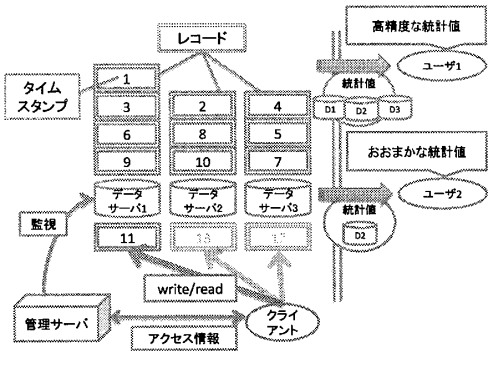

図 1　システム構成

† 日本電信電話 (株), NTT　‡ 早稲田大学, Waseda Univ.

[*1] 計算サーバをデータサーバと独立に持つことや，計算結果ではなくサンプルそのものを返すことも可能である．

御できる．そして，高い精度，すなわち大きいサンプルを求める場合は対象データサーバ数が多くなり，低い精度で良い場合は少なくなるので，ストレージ全体へかかる負荷と要求精度とのトレードオフが成立する．

3.2 規模拡張性

各データサーバは通常のデータベースであり，個々におけるトランザクションは原則として ACID 特性 [5] を満たす．しかし，ストレージ全体としては，ACID の D(Durability:永続性) は保証せず，統計的性質の保持のみを保証すればよい．そしてその統計的性質の保持は，ランダムサンプリング機能による各データサーバのデータの統計的等質性により自然に実現される．従って，各データサーバは基本的に自身へのアクセスに対応するだけでよく，2 相コミットのように，お互いに同期して分散トランザクションを管理する必要はない．つまり，各データサーバが独立に処理を進めていても，ストレージ全体として不整合が生じるようなことはない．

データサーバを追加する際は，各データサーバの独立性から，既存のデータサーバへ影響を与えることはなく，追加後にそのデータサーバに蓄積されるデータも統計的等質性を保てる．つまり，既存データサーバからデータを移動する必要もない．但し，稼動期間の管理とその偏りの処理は必要になる*2．

以上のように，既存の分散ストレージに比べ，分散トランザクションが不要であること，メンバシップ管理が簡素であることから，システム全体として高い規模拡張性が期待できる．

3.3 障害耐性と可用性

各データサーバのデータの統計的均質性により，一部のデータサーバに障害が発生しても，それに伴うデータの消失による全体として特定の偏りは発生しない．また同様に，消失したデータを復元せずとも，他のデータサーバのデータで代用することができる．これはレコードの書き込みでも同じである．データサーバが一時的にダウンし，それを管理サーバが見逃がしたとする．その機会はデータサーバ間で均等に起こり得るものであるから，書き込まれるはずだったいくつかのレコードは失われるが，全体として偏った消失にはならない．従って，データサーバ障害の影響は最高精度の低下のみで済み，その度合いもデータサーバ数が十分多ければ問題にはならない．よって，冗長構成を取る必要もなく，高い障害耐性および可用性が期待できる．

3.4 管理サーバ

管理サーバにはクライアントのアクセスが集中するが，データそのものは扱わず，かつ監視対象のデータサーバ自身には影響を及ぼさないことから，既存技術での分散化も可能である．また，2.3 節で述べたように，クライアントにデータサーバ選択のロジックを移すことや，サーバリストをキャッシュさせることで，管理サーバへのアクセスを大きく減らすことも可能である．

以上から，可用性やレイテンシを下げることはないと考えている．

4 サンプルサイズとデータサーバ数

2.3 節で述べたサンプルサイズとデータサーバ数の決定について，母集団のサイズ N が十分大きい状況での平均値 μ の区間推定を例として，以下説明する．なお，母比率の推定や，他の統計量についてもほぼ同様の議論が可能である．

4.1 サンプルサイズ

要求される精度として，信頼率 $1-\alpha$ と許容する誤差 e が与えられているとする．この時の最小サンプルサイズを n，母分散を σ^2 とすると，サンプル平均 u の信頼区間は，

$$u - z_\alpha \sqrt{\sigma^2/n} \leq \mu \leq u + z_\alpha \sqrt{\sigma^2/n}$$

である．但し，z_α は標準正規分布の両側 $100\alpha\%$ 点である．この幅が $2e$ に等しいとして n について解くと，

$$n = z_\alpha^2 \sigma^2 / e^2$$

となる．σ^2 が得られない場合は，N, n が十分大きいという前提の下，サンプルの不偏分散 V を用いることができる．

4.2 データサーバ数

稼動している全データサーバ数 M に対し，参照するデータサーバ数 m を決定するためには，サンプルサイズに加えてデータサーバあたりの検索レコード数 r が必要となるが，これはデータサーバ毎に異なり，実際に検索してみなければわからない．そこで，まず k ($k < M$) 台のデータサーバに検索をかけ，そのレコード数 r_k を用いて $r = r_k/k$ と見做す．同様に全検索結果の不偏分散 V_k を求め，$V = V_k$ と見做す．すると，必要なデータサーバ数 $m = \lceil n/r \rceil$ となるので，残りの $m-k$ 台に対して検索を改めてかければ良い．

なお，r ならびに V の不確かさがレコード数を不足させる可能性がある．得られたレコード数を検索後に確認し必要に応じて再度検索をかけるという方法もあるが，実際の運用上は，m を大きめに取ることで対処できると考えている．

また，初回の検索によるオーバヘッドを小さくするために，保存データを故意に粗にしたデータサーバを，見積用として他のデータサーバと別に運用することもできる．クライアントからの書き込みを受けたデータサーバが $1/h$ ($h \gg M$) の確率で見積サーバに複製を残しているとする．初回の検索を見積サーバに対して行い，そのレコード数と不偏分散をそれぞれ r_0，V_0 とした時，$r = h/M \cdot r_0$，$V = V_0$ と見做す．あとは $m = \lceil n/r \rceil$ 台に検索をかければ良い．全レコード数が十分大きく，M があまり大きくない場合はオーバヘッドの削減に有効であると考えている．

5 おわりに

センサデータの統計的利用を前提とした分散サンプリングストレージのアーキテクチャを提案した．これに基づくシステムは，高い規模拡張性と可用性を持ち，ネイティヴなサンプリング機能と障害耐性を備えている．さらに，統計処理の精度を確率的に保証する枠組みを持つことを特長とする．今後，プロトタイプの作成を行い，詳細に定式化した精度保証の枠組みに沿って評価実験を行う予定である．

参考文献

[1] “Internet of Things in 2020: roadmap for the future,” European Technology Platform on Smart Systems Integration final report, May 2008.

[2] M. Balazinska, A. Deshpande et al., “Data management in the worldwide sensor web,” IEEE Pervasive Computing, vol.6, no.2, pp.30–40, Apr. 2007.

[3] 有村博紀, “大規模データストリームのためのマイニング技術の動向,” 信学論（D）, vol.J88-D-I, no.3, pp.563–575, Mar. 2005.

[4] 佐藤浩史, 井上武 他, “センサデータマイニングのための分散サンプリングストレージの提案～統計量の基本的な挙動解析を中心に,” 信学技報, vol. 109, no. 449, IN2009-183, pp. 235–240, 2010 年 3 月.

[5] T. Haerder and A. Reuter, “Principles of transaction-oriented database recovery,” ACM Comput. Surv. 15, 4, pp.287–317, Dec. 1983.

*2 この詳細は別稿に譲る．