CBoCによる大規模分散処理システムの実現

高倉 健（たかくら たけし）/空 一弘（そら かずひろ）
天海 良治（あまがい よしじ）/鷲坂 光一（わしさか みつかず）
冨田 清次（とみた せいじ）

NTT情報流通プラットフォーム研究所

データマイニングのように大量データの処理により新しい付加価値情報を創出するサービスの実現には，大量データを多数のPCサーバを用いて効率的に分散処理する大規模分散システムが必要となります．本稿では，CBoC（Common IT Bases over Cloud Computing）による大規模分散処理システムの中核をなす分散ファイルシステム・分散テーブル・分散ロックについて紹介します．

はじめに

クラウドコンピューティングの適用領域の1つとして，データマイニングなどで利用される大規模データ処理の分野があります．データマイニングはシステムに格納されている大量データを処理することにより付加価値情報を創出するサービスで，具体例としては，Webコンテンツを集めて分析を行う検索サービスや，ログ情報などを分析してユーザの嗜好に合致した情報を提示するレコメンデーションサービス等が挙げられます．

これらのサービスで取り扱うWebコンテンツやログ情報は，数十TB～数百TB（テラバイト），場合によってはそれ以上の規模になります．このため，大量データを多数のPCサーバを用いて効率的に分散処理するアーキテクチャが注目されています[(1)～(4)]．

NTT情報流通プラットフォーム研究所ではクラウドコンピューティング共通技術CBoCの取り組みの中で，多数のPCサーバによる分散環境を用いて大量データの処理を実現する「大規模分散処理システム」の技術開発を進めています．本稿では，大規模分散処理システムの果たす役割と中核となる3つの技術について説明します．

大規模分散処理システム

図1に示したログ情報分析の例で

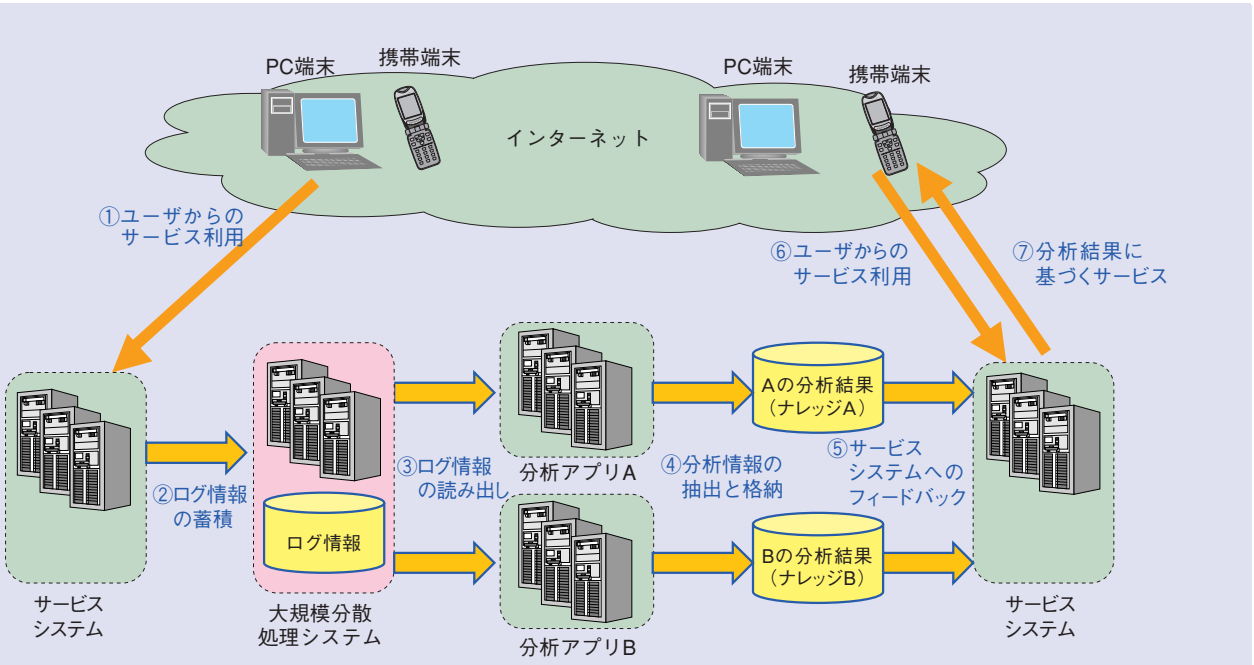


図1　大規模分散処理システムのログ情報分析への適用例

は，ネットワーク上に多数のサービスが構築され，ユーザのサービス利用履歴がログ情報として大規模分散処理システムに蓄えられます．分析等のアプリケーションはさまざまな観点からログ情報を処理して分析結果をナレッジとして保持します．サービスシステムがナレッジを活用する仕組みを設けることで，分析結果に基づくユーザ特化型のサービスが提供可能となります．

この例のようにレコメンデーションサービスや検索サービスでは，システムに書き込まれた大量のデータを，サービスに応じたさまざまな形態で読み出して処理するため，参照系のデータアクセスが中心となります．大量データの高速処理という特定用途の実現のために，大規模分散処理システムでは参照系データアクセスを重視することにより，性能と容量を追求した分散データベース機能を実現します．また，分散処理を有効に機能させるために，大規模分散処理システムにはマシン台数を増加させれば取扱い可能なデータ容量も増加できるスケールアウト性が必要となります．さらに，多数のPCサーバによるシステム構成では，1台のPCサーバの障害がシステム全体に波及してはいけないので，耐障害性が重要になります．

本分野への関心が高まり，コミュニティとして大規模分散処理システムを開発する動きが出てきました．著名な例としてApacheプロジェクトのHadoop [(5)] がありますが，ここでは信頼性・耐障害性よりも機能・性能面の取り組みが重視されています [(6)]．これに対してCBoCでは，障害に強い大規模分散処理システムの実現を目指し技術開発を進めています．

システム構成

CBoCによる大規模分散処理システムは**図2**に示すように，大量データを多数のPCサーバに分散して格納するための分散ファイルシステム，大量データを構造化データとして管理する分散テーブル，これらの分散システムの可用性とシステムとしての一貫性を高めるための基本機能を提供する分散ロック，から構成されます．そしてさまざまなアプリケーションから直接，あるいは分散処理ライブラリを介して利用されます．以下では，大規模分散処理システムの3つの技術について説明します．

(1) 分散ファイルシステム

分散ファイルシステムは，大量のデータを確実に保持し，必要なデータを提供することで大規模分散処理システムを土台として支える役割を果たします．**図3**に示すように，分散ファイルシステムは，全体のデータ管理と制御を司るマスタサーバ，物理的にデータを保持するワーカ，分散ファイルシステムを利用するアプリケーションにリンクされるクライアントライブラリの3つの要素から構成されます．

検索サービスの検索インデックス生成やデータ分析サービスの分析処理では，収集した大量のデータを読み出してデータ処理が実行されます．この処理を短時間で実行するには，高いスループットとデータ処理能力が発揮できる分散ファイルシステムが必要とな

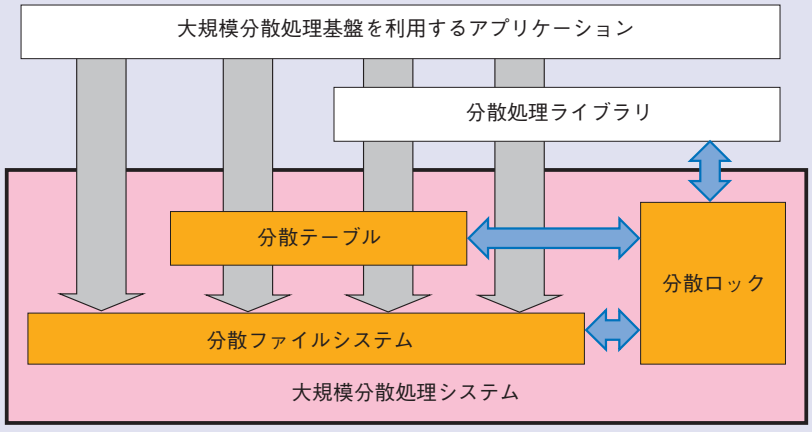


図2　大規模分散処理システムの構成

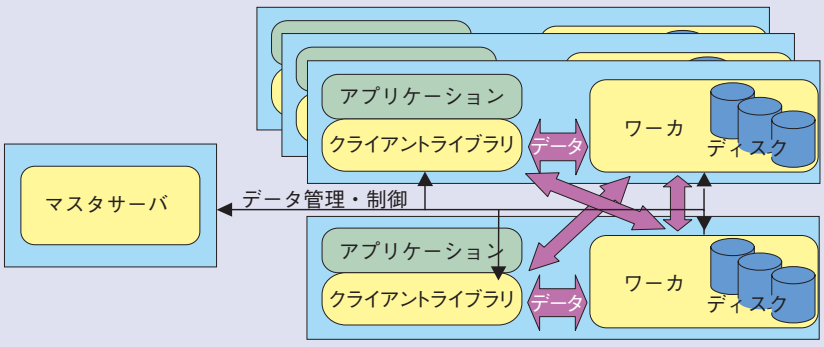


図3　分散ファイルシステムの構成

ります．

単に数十TBのデータを保持するだけなら従来のユニット型ストレージ装置でも可能ですが，多数のサーバに分散して格納することで以下の利点が得られます．

・高スループット：多数のサーバに分散してデータの書き込み，読み出し処理を実施します．リクエストの分散と結果の集約に要するオーバーヘッドを抑制することで，高スループットが得られます．

・耐障害性：データの複製を物理的に異なる複数のサーバに保持することで，単一機器に故障が生じてもシステム全体の停止につながることのない構成を取ることができます．

・拡張性：サーバを追加できれば，分散ファイルシステム全体が保持するデータ容量を増やすことができます．サーバの追加処理を容易にすることで拡張性を高めています．

・高いデータ処理能力：データを保持しているサーバ上でデータ処理を実行することにより，ストレージからCPUへのデータ移動を最小限に抑制でき，台数分のCPUを有効活用して処理することが可能になります．

⑵　分散テーブル

分散テーブルは，大量データを構造化データとして分散サーバ上で管理する役割を果たします．通常の分散ファイルシステムの場合，大量データの単純な読み書きには適していますが，ランダムアクセス機能やデータ検索機能といった，セマンティックスに基づくデータアクセス機能やデータ管理機能を十分には備えていません．分散テーブルは，大量データを構造化データとして使えるよう，用途を絞りつつ可能な限り効率化・高速化し，簡素化されたデータベース管理システムとして機能します．

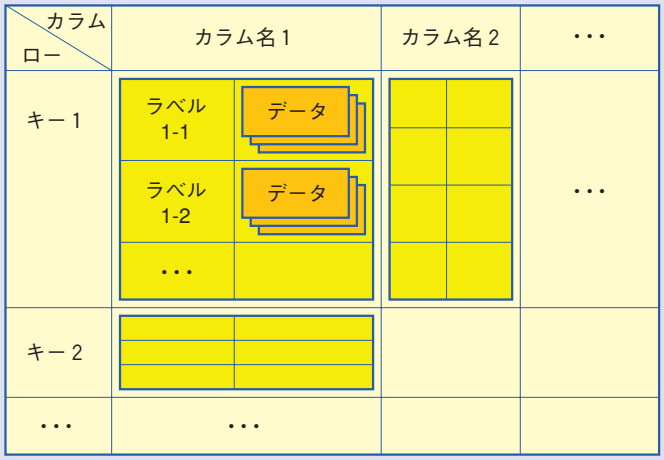


図4　分散テーブルにおけるデータモデル

分散テーブルで管理するデータモデルについて，**図4**を用いて説明します．分散テーブルでは，複数のカラムからなるデータをローキーの値順に整列させて保管します．これにより，キー値に基づくデータへのアクセスが簡単になり，キー値による範囲検索を高速に処理できます．また，分散テーブルではラベル付けされた複数のデータを管理することができ，ラベルに対応するデータ群として複数世代のデータを管理することができます．これにより格納データの最近の変遷を保持するとともに，不要になった古いデータを自動的に削除することができます．

分散テーブルでは，テーブル情報を多数のPCサーバで分散管理することが可能です．また，データ量の増加に応じてPCサーバを動的に増加させることも可能です．これによって，数百台のPCサーバに分散して百TB規模の大規模データを扱えるスケーラビリティを実現します．

分散テーブルは，テーブル情報を多数の構造化されたファイルに分割し，分散ファイルシステムを用いて管理します．分散ファイルシステムの持つ耐障害性，可用性，拡張性を活用することで，大量のテーブル情報を確実に保持することができます．分散テーブルの性能を十分に引き出すには，分散ファイルシステムの処理能力が発揮できるよう，両システムの連携が重要です．例えば，分散環境の台数分のPCサーバのCPUを有効活用するために，分散テーブルが動作するデータ管理元のPCサーバと，分散ファイルシステムがデータを物理的に格納するPCサーバとを，可能な限り同一にすることで，サーバ間の通信オーバーヘッドを削減でき，性能向上に結び付けています．

⑶　分散ロック

分散システムを構築するためには，分散して処理を行うサーバ間で資源の共有や資源への排他的なアクセスをどのように実現するのか，また，サーバ故障やネットワーク分断の際に，どのようにしてシステムの可用性や一貫性を保つのかが大きな問題になります．分散ロックは，これらの問題を解決するための機能を提供します．また，分散ロック自体も複数サーバで構成され，サーバ間でファイルの複製を持つ，高

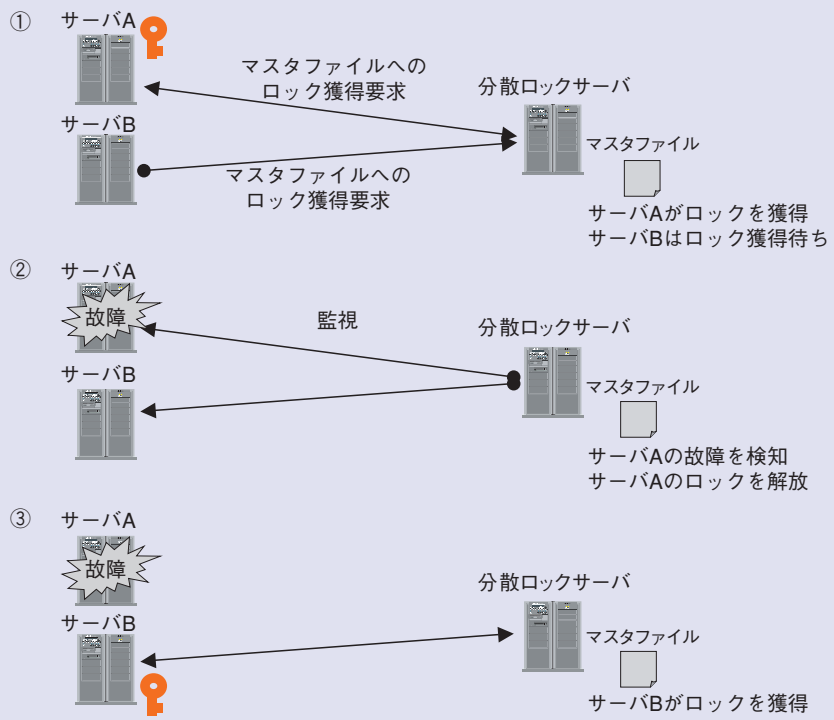


図5　分散ロックを利用したマスタ切替

可用な構成を取っています．

分散ロックは，サーバ間の情報共有のための小容量データ向き共有ファイルシステム，サーバ間の排他制御をファイルを利用して実施する排他ロック，ファイル操作やロック操作に関連したイベント通知，の3つの機能を提供します．

分散ロックを利用して，2台のサーバ間でマスタ機能を切り替える手順を**図5**に示します．

① サーバA，サーバBは分散ロックサーバへ同一ファイルへのロック獲得要求を送ります．分散ロックサーバは，要求を直列化することで，1つのファイルに関しては高々1つのサーバがロックを獲得することを保証しています．この場合，ロックを獲得できたサーバAがマスタサーバとして動作します．

② 分散ロックサーバはサーバAと，定期的にKeepAlive情報を通信することで，サーバAの死活状態を監視しています．サーバAの故障を検知すると，サーバAが獲得していたロックを解放します．

③ ロック獲得待ちしていたサーバBが新たにロックを獲得し，サーバBがマスタサーバとして動作します．

このようにして，高可用なマスタ構成を実現し，複数のマスタサーバが同時に動作するなどの資源への破壊的なアクセスにより，システムの一貫性が壊れることを防止しています．

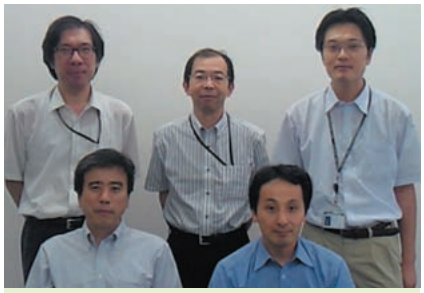
今後の展開

クラウドコンピューティングへの注目が集まり，大量データを活用することの重要性が，改めて認識されてきています．これに伴い，大量データの処理が可能な情報処理システムの必要性も高まりつつあります．

本稿では，CBoCによる大規模分散処理システムの実現に向け，中核となる3つの技術を紹介しました．今後も，耐障害性と信頼性を確保し，高速で使いやすいシステムとなるよう性能向上と機能拡充を図り，大量データ処理を用いた付加価値サービスの創出に貢献していきたいと考えています．

■参考文献

(1) S. Ghemawat, H. Gobioff, and S.T. Leung：“The Google File System，” Proceedings of the 19th ACM Symposium on Operating Systems Principles，pp.20-43，2003.
(2) M. Cafarella, E. Chang, A. Fikes, A. Halevy, W. Hsieh, A. Lerner, J. Madhavan, and S. Muthukrishnan：“Data Management Projects at Google，” SIGMOD Record, Vol.37，No.1，pp.34-38，March 2008.
(3) F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach， M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber：“Bigtable: A Distributed Storage System for Structured Data，” OSDI，2006.
(4) M.Burrows：“The Chubby lock service for loosely-coupled distributed systems，” 7th USENIX Symposium on Operating Systems Design and Implementation（OSDI ’06），2006.
(5) http://hadoop.apache.org/
(6) http://preferred.jp/pub/hadoop.pdf



（後列左から）天海　良治/ 鷲坂　光一/
空　　一弘
（前列左から）冨田　清次/ 高倉　　健

クラウドコンピューティングによる新たなサービスの実現に向け，実用化技術の開発を通して，皆様のご意見・ご要望にこたえられるよう取り組みを進めていきます．

◆問い合わせ先
NTT情報流通プラットフォーム研究所
ソフトウェア基盤推進プロジェクト
TEL　0422-59-6009
FAX　0422-59-3739
E-mail　cboc@lab.ntt.co.jp