データベースにおけるリアルタイム構造劣化監視機構の試作

Prototype of Real-time Structural Deterioration Monitor for DBMS

星野 喬♠ 合田 和生♦ 喜連川 優♦

Takashi HOSHINO Kazuo GODA
Masaru KITSUREGAWA

近年，データベース管理コスト削減，及び，データベースの常時運用が求められている．構造劣化は，データベース更新が繰り返されることで生じるデータ格納構造の非効率性であり，性能低下を引き起こす．データベース再編成は構造劣化を除去するために不可欠な管理業務である．再編成を行う時間と対象を決定するために，構造劣化の監視が必要であるが，これまでは構造劣化と性能低下の因果関係が明らかになっておらず，管理者の経験則による効率的とは言えない再編成判断が行われてきた．再編成判断の容易化は，データベース管理コスト削減に貢献する．本論文では，再編成判断を容易化する構造劣化モデルと常時運用に対応したリアルタイム監視手法を用いたオンライン構造劣化モニタを試作し，TPC-C ベンチマークを用いて評価を行った．

Nowadays, reduction of database administration cost and full-time database operation are required. Structural deterioration, which is disorganization of data structure caused by database updates, leads to performance degradation. Database reorganization is essential to remove structural deterioration. In order to determine reorganization target and time, monitoring of structural deterioration is required. However, it still has not been clear how structural deterioration leads to performance degradation, therefore database administrator makes decisions about reorganization inefficiently. Facilitating reorganization decision contributes reduction of database administration cost. In this paper, we made a prototype of on-line structural deterioration monitor with a structural deterioration model to facilitate reorganization decision and with a real-time monitoring scheme for full-time database operation, and we evaluated it with TPC-C benchmark.

1. はじめに

近年，様々なITシステムにおいて運用されているデータベースが大容量化している．また，データベースの用途が拡大するにつれ，元々複雑に階層化したデータベースシステムは構成要素や設定パラメータが増えるなど，複雑さを増している．データベース管理者一人あたりの管理できる容量，複雑性には限界があるため，データベース管理コストが高くなり，それを削減する技術が求められている．また，データベースの常時運用も必要とされており，運用中のトランザクション等に影響を与えないように状態の監視や管理業務の実行を行うことも求められている．このため，データベース管理の自立化を目指す試みが行われている[2].

我々は，データベース管理業務の中で容易化，自立化が難しい再編成業務に着目した．データベース更新操作が繰り返されることによりデータ格納構造は非効率化する．これを構造劣化と呼ぶ．構造劣化は性能を大幅に低下させる場合があるため，管理者は再編成を行うことにより，データを再配置し，性能低下を防ぐ必要がある．再編成業務の実施には，再編成判断，即ち，再編成を行う対象と時間を決定することが必要であるが，これまではデータベース状態を示すいくつかのパラメータを参考に管理者が経験則による再編成判断を行っていた．この方法は構造劣化と性能低下の因果関係が明確ではないまま行われていたため，管理者が性能低下の原因を構造劣化と断定できぬままに再編成を実施し性能が回復しない場合など，非効率な再編成が多く行われてきた．構造劣化監視を行うことで，再編成判断を容易化，効率化できる．また，再編成判断を自動化すれば，再編成業務全体の自立化を実現に近づけることができる．これらの取り組みにより，データベース管理コストを削減することが期待できる．

我々は，構造劣化を引き起こすデータ配置の性質をモデル化し，構造劣化の定量的表現を可能にした．特に，ストレージ性能特性が性能低下量に大きく影響を与える構造劣化であるデータのクラスタ化度低下について，再編成判断を適切に行うために，構造劣化に起因する性能低下量を範囲検索IOコストによって推定する手法を提案した．また，常時運用に対応した監視を実現するために，運用中のデータベーストランザクションへの影響を最小限に抑える構造劣化監視方式を提案した[6]. 本論文では構造劣化モニタ試作機を実データベースシステムに実装し，構造劣化をリアルタイムに観測できることを確認した．

本論文は次のように構成される．まず，第2章で関連研究について述べ，次に第3章で構造劣化モデルとオンライン構造劣化モニタ試作機について述べる．その後，第4章で評価を行い，最後に第5章にて結論と今後の課題を述べる．

2. 関連研究

これまで，データベース再編成に関する研究は，オンライン再編成方式など実行方式の高度化[4]，及び，実行スケジューリングの最適化を中心に行われてきた[1]. 再編成判断は後者にあたり，これらの研究では，性能低下量を予測し，運用時間あたりの再編成コストを最小化するスケジューリング手法が提案された．

性能推定に関連して，文献[3]はデータベースバッファの影響を考慮して，索引と表を走査するのに必要な IO 数をモデル化した．この文献はデータがクラスタ化されていない状態を仮定している．我々の研究はストレージ性能特性を考慮しており，クラスタ化の度合いに対して大きく非線形に性能が変化する現象に対応できる．

文献[5]において，部分再編成手法が提案された．それまで再編成対象となる最小粒度は，表単位もしくは表空間単位であったが，この文献は大容量の表に対して，構造劣化部分のみを再編成することにより，再編成時間を削減しつつ，構

♠ 学生会員 東京大学大学院情報理工学系研究科
hoshino@tkl.iis.u-tokyo.ac.jp
♦ 正会員 東京大学生産技術研究所
{kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

造劣化をほぼ回復することが可能であることを示した．この文献では，更新パタンについて 2 つのケーススタディを挙げるに留めているが，本研究では，データ配置から直接構造劣化による性能低下量を推定できるため，更新パタンに依存しない構造劣化の監視が可能である．また，本研究は細粒度での構造劣化部分の同定を容易にし，部分再編成対象の効率的な決定に貢献することが期待される．

3. 構造劣化監視機構

本研究が目指す自立再編成機構の枠組みと，その中での構造劣化モニタの位置付け，及び，構造劣化モデルの詳細については文献[6]に述べた．本章では構造劣化モデルについて簡単にまとめた後，構造劣化モニタ試作機の実装方式について述べる．

3.1 構造劣化モデル

本節では，構造劣化モデルについて述べる．表 1 に本節で用いる変数，定数及び関数の一覧を示した．ここでは，クラスタ表を構成する B+木構造を対象に議論を進める．このような構造は MySQL のクラスタ表や Oracle の索引構成表で使われている．クラスタ表の B+木構造は，枝及び根ページにはクラスタ鍵(Cluster Key)と下位ページへのポインタが，葉ページにはデータ行が格納されている(図 1)．葉ページは，論理空間ではクラスタ鍵順に並んでおり，この順に番号 i をつける．各ページ i はストレージ上の位置であるページアドレス $\mathrm{p}[i]$ を持っている．葉ページ列アドレス全体を $\mathbf{P}=\{p[i]\mid 0\le i<N\}$ と定義する．範囲検索は対象となる範囲(例えば図 1 における範囲 $\mathbf{R}$）に属する連続部分葉ページ列を走査する．$\mathbf{R}$ によって走査対象の部分葉ページ列は一意に決まる．これを $[i_0,i_1]$ とする．一般的には，B+木における葉ページの論理順序とストレージ内の物理順序が対応しており，範囲検索時にシーケンシャルアクセスとなることが期待されている．これをクラスタ化された状態と呼ぶ．データベース更新によって，B+木の構造が変化すると，各ページの物理的な配置が論理順序と物理順序の対応が乱雑化し，クラスタ化の度合いが低下する．このとき範囲検索の IO は非シーケンシャル化し，性能が大幅に低下する．範囲検索は同じページを一度しか読まないため，バッファキャッシュヒットを期待しづらく，一般的に IO ボトルネックになる．このとき，範囲検索応答時間は対象の葉ページ列走査の IO 時間がほとんどを占める．そのため，範囲検索において，個々の葉ページ i を読み込む IO コスト $c[i]$ は，$p[i]$ と直前に読み込んだページアドレス $p[i-1]$ から，ストレージ性能モデル[6]を現す関数 f を用いて以下の式(1)で推定できる．

$$c[i]=\begin{cases} C_0 & i=0 \\ f(p[i]-p[i-1]) & i>0 \end{cases} \qquad (1)$$

このとき，範囲 $\mathbf{R}$ に対する範囲検索 IO コスト $C_{\mathbf{R}}$ は，式(2)で近似される．これを範囲 $\mathbf{R}$ の構造劣化量とする．

$$C_{\mathrm{R}}=\sum_{i=i_0}^{i_1} c[i] \qquad (2)$$

C_0 は，シーケンシャル読み込みにおけるページの IO コストを表すものとする．ディスクドライブから構成されるストレージにおいて，シーケンシャルアクセス時の IO コストが最小であることは明らかであるため，$c[i]$ の取りうる値の中で C_0 が最小値とする．$\mathbf{C}=\{c[i]\mid 0\le i<N\}$ を $\mathbf{P}$ に対する構造劣化分布とする．

表 1　変数，定数及び関数一覧表
Table 1　Variables, constants and functions

変数	意味
$\mathbf{R}$	クラスタ鍵範囲
$p[i]$	ページ i のアドレス
$\mathbf{P}$	葉ページ列アドレス列
f	ページ列から IO コストを導く関数
$c[i]$	範囲検索におけるページ i の読み込み IO コスト
C_0	ページ読み込み最小 IO コスト
$\mathbf{C}$	$\mathbf{P}$ に対する IO コスト列(構造劣化分布)
$C_{\mathbf{R}}$	範囲 $\mathbf{R}$ の範囲検索 IO コスト

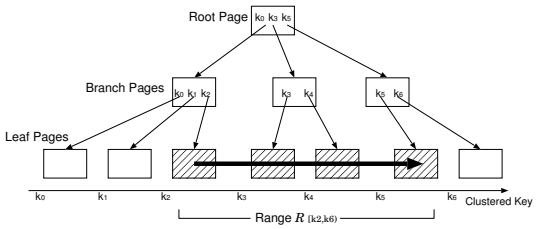


図 1　クラスタ表の B+木構造
Fig. 1　B+Tree structure of clustered table

データベース更新による B+木の構造変化は，ページの分割もしくは結合である．$\mathbf{P}$ に対してはページが追加されるか，削除されるかのどちらかである．このとき，$\mathbf{P}$，$\mathbf{C}$ の変化は対象ページのごく近傍のみで起こるため，$C_{\mathbf{R}}$ も含め，差分計算によって最新のデータベース状態に対する推定値を保持することが出来る．全範囲を対象に $\mathbf{C}$ を保持すると，式(2)を用いて任意の範囲 $\mathbf{R}$ の構造劣化量 $C_{\mathbf{R}}$ が推定できる．

3.2 構造劣化モニタ試作機

本節では，試作した構造劣化モニタについて述べる．試作機は，MySQL 5.0 InnoDB データベースにおけるクラスタ表のクラスタ鍵および二次索引の鍵を論理軸として，構造劣化分布を描画できる．モニタ対象となる表，索引構造は B+木で実装されている．図 2 に試作機の構成を示した．試作機は内部データとしてメモリ上に B+木構造およびストレージ性能特性モデルを持ち，プログラムは主に初期化ルーチン，B+木更新差分適用スレッド，構造劣化分布描画スレッドから構成される．

試作機を起動すると，初期化ルーチンがデータベースのディクショナリから存在する表および索引のスキーマを認識す，ストレージ性能特性モデルパラメータを設定ファイルから読み込む．次に監視対象の表(索引)を構成する B+木データ構造を読み込み，その時点における構造劣化分布を計算し，メインメモリ内に B+木構造を作成して保持する．その後，B+木更新差分適用スレッドが起動し，更新差分の取得待ちになる．また，構造劣化分布描画スレッドが起動し，構造劣化分布が更新される度，もしくは定期的に描画を行う．B+木更新差分適用スレッド及び構造劣化描画スレッドは適宜排他を取って内部 B+木構造を更新，参照する．

データベース運用中，B+木の構造が変化する更新，即ち，ページの追加，削除が起きる度に，データベースエンジンに実装された B+木更新差分抽出器から共有メモリに作成されたキューを介して更新差分を抽出する．更新差分は追加(削除)対象ページとその論理空間における左右のページのクラスタ鍵，ページアドレス，表(索引)番号から成る．抽出した

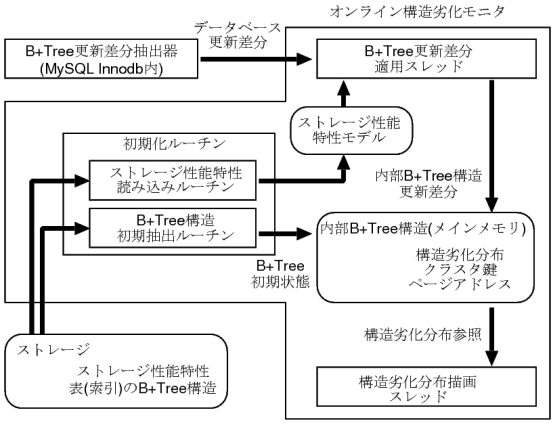


図 2　構造劣化モニタ試作機の構成
Fig. 2　Construction of S.D. monitor prototype

更新差分は適宜 B+木更新差分適用スレッドに渡され，内部 B+木構造をインクリメンタル更新することでほぼ最新のデータベース状態に追随する．B+木更新差分抽出機は，データベースエンジン内に実装されているため，対象ページが処理のためにメインメモリに読み込まれているときに更新差分を取得する．このとき，情報抽出のためのストレージへの実 IO はほとんど発生しないため，運用中のトランザクション等に対する影響を最小限にすることができる．

構造劣化分布描画スレッドは，各表，索引毎に，論理軸および物理軸における構造劣化分布 $\mathbf{C}$ を，C_0 を基準に正規化して表示する．論理軸とは，クラスタ鍵順に対応する B+木の葉ページ列の順序 i である．物理軸とは，各葉ページのページアドレス $p[i]$ の順序である．論理空間と物理空間におけるデータ分布の対応を分かりやすくするため，クラスタ鍵空間を区間に分けてデータを色分け表示している．

4. 評価

本章では，構造劣化モニタ試作機が構造劣化分布を十分な精度，粒度で推定しているか，リアルタイムに更新差分を反映できるかについて評価を行った．

実験環境として，Linux が動作する PC 上で，ディスクドライブ 1 台を raw デバイスとして用い，MySQL 5.0 InnoDB データベース表空間 1GB を作成した．ページサイズは 16KB とし，クラスタ表を用いて TPC-C スキーマを作成した．使用ドライブの最外周ゾーンにおける 16KB シーケンシャルリード最大スループットは 28MB/s であった．このときのページあたりの応答時間を C_0 として用いた．ウェアハウス数は 2 として初期データをロードした．

TPC-C は卸売り会社のトランザクション処理を模擬したベンチマークであり，注文(new-order)，支払(payment)，配達(delivery)，注文確認(order-status)，在庫確認(stock-level)の計 5 トランザクションが定義されている．純正 TPC-C では order, order-line, history 表のデータが単調増加するが，実運用システムでの長期的運用を考慮し，上記のデータ量がほぼ一定になるような移送処理を追加した(move-old-orders, move-old-history)．これら計 7 トランザクションが 40 プロセスから並列に計 100000 トランザクションを投入する操作を 1 loop とし，計 3 loop 実行した．各プロセスは思考時間 0 で new-order: 45%，payment: 45%，delivery: 6%，order-status: 1%，stock-level: 1%，move-old-orders:1%，move-old-history: 1% の割合でランダムに投入した．更新対

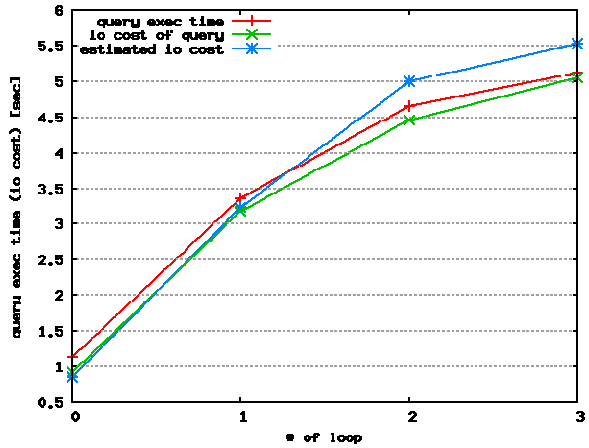


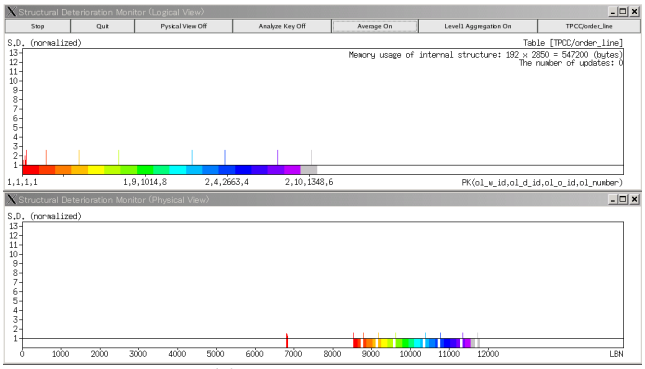
図 3　order_line 表レンジスキャン応答時間と IO コストの測定値と推定値
Fig. 3　Response time of range scan of order_line table, and measured and estimated IO cost

象のウェアハウスは 1 つのみとした．また，純正 TPC-C には定義されていない集計処理 promising_sales を定義した．これは対象ウェアハウスにおける配達待ち注文による売り上げ期待額を集計するもので，order_line 表の範囲検索を伴う．当該集計処理において，データのクラスタ化度低下による性能低下が発生することが予測される．本実験では，各 loop の前後に更新ウェアハウスを対象に promising_sales クエリを発行し，応答時間を測定した．同時に IO をカーネルレベルでモニタし，応答時間に占める IO コストを算出した．また，構造劣化モデルによる推定 IO コストを算出し，比較した．

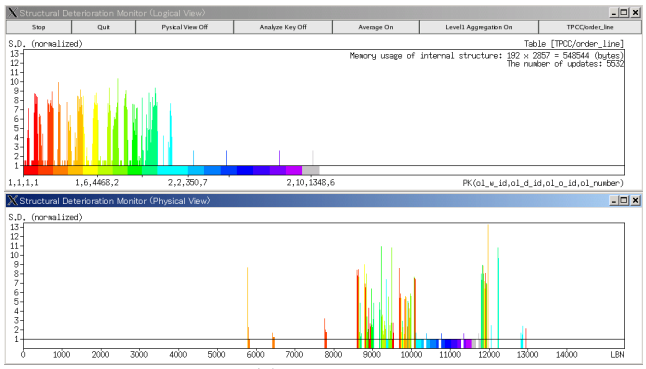
図 3 に promising_sales クエリの応答時間とその IO コスト，及び，構造劣化モデルによる IO コスト推定値を示した．クエリ応答時間と IO コストの差は CPU によるデータ処理時間と考えられ，応答時間のほとんどを範囲検索 IO コストが占めていることが確認できる．モデルによる IO コスト推定誤差は 6～12%であり，性能と比較可能な形でクラスタ化度を表す構造劣化量をモデル化できたと言える．

図 4 に構造劣化モニタによる order_line 表の構造劣化分布のトランザクション loop 実行に伴う変化を示した．論理軸において更新ウェアハウスに相当する部分の構造劣化量が loop 実行に伴って増加していくのが確認できる．物理軸においては，構造劣化しているデータは，ストレージ内に分散している様子が確認できる．また，更新分布が偏っているため，構造劣化している部分とそうでない部分が交互に発生していることも確認できる．このような構造劣化分布の偏りをクエリ応答時間計測によって得るためには，様々な範囲において範囲検索クエリを実施しなければならず，非現実的である．本試作機による細粒度の構造劣化分布把握を行うことで，構造劣化部分同定の高速化が期待できる．

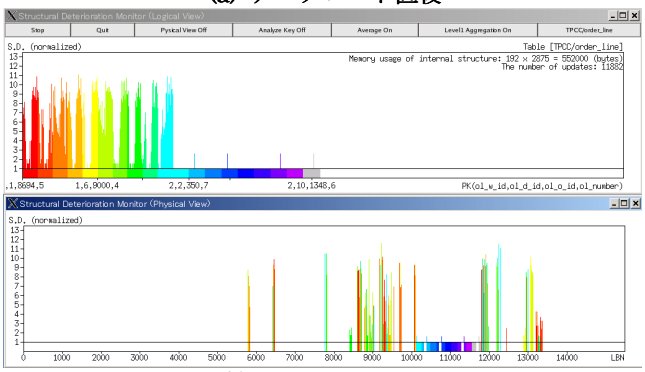
次に，作成した構造劣化モニタ試作機がリアルタイムに構造劣化をモニタできるか評価した．試作機を稼動したコンピュータにおいて，抽出した更新差分を事前に蓄積しておき，高速に適用することにより，168.4 更新差分数/秒で処理できることを確認した．先の実験において更新発生頻度は 13.7 更新差分数/秒であったため，12 倍以上高速に更新差分を適用できることが分かった．本試作機は 1 秒間隔で描画を行っており，1 秒前のデータベース状態に追随できるため，ほぼリアルタイムに構造劣化分布を監視できるといえる．



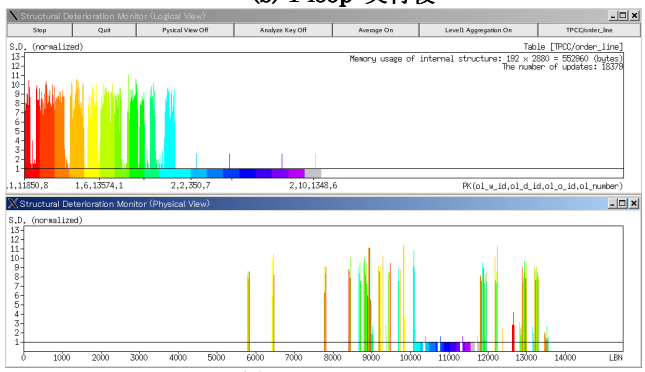
(a) データロード直後



(b) 1 loop 実行後



(c) 2 loop 実行後



(d) 3 loop 実行後

図 4 TPC-C ベンチマークにおける order_line 表の構造劣化モニタ試作機による表示

Fig. 4 View of order_line table on TPC-C benchmark by S.D. monitor prototype

5. おわりに

本論文は、データベースにおけるリアルタイム構造劣化モニタを試作し，評価した．構造劣化モニタが高い精度と粒度で構造劣化を監視できることを示し，ほぼリアルタイムに最新のデータベース状態の構造劣化分布推定が可能であることが示された．

今後の課題として，より包括的な構造劣化のモデル化，再編成スケジュール方式の構築を経て，再編成を自立化することが挙げられる．

[謝辞]

本研究の一部は，文部科学省リーディングプロジェクト e-society 基盤ソフトウェアの総合開発「先進的なストレージ技術」の助成により行われた．協力企業である株式会社日立製作所より多くの有益なコメントを頂戴した．深謝する次第である．

[文献]

[1] Batory, D. S.: “Optimal file designs and reorganization points”. ACM TODS 7(1), pp.60-81 (1982).

[2] Lightstone, S., Schiefer, B., Zilio, D. and Kleewein, J.: “Autonomic Computing for Relational Databases: The Ten Year Vision”. In Proc. of AUCOPA2003 (2003).

[3] Mackert, L. F., and Lohman, G. M.: “Index Scans Using a Finite LRU Buffer: A Validated I/O Model”. ACM TODS 14(3), pp.401–424 (1989).

[4] 合田和生，喜連川優.：“データベース再編成機構を有するストレージシステム” 情報処理学会論文誌データベース，46(TOD (SIG8)) pp.130-147 (2005).

[5] 合田和生，喜連川優.：“構造劣化の局所性を活かしたデータベース部分再編成の提案” DBSJ Letters，5(1), pp.109-112 (2006).

[6] 星野喬，合田和生，喜連川優．：“データベース更新差分を用いた範囲検索の IO コスト推定” DBSJ Letters，4(2)，pp.37-40 (2005).

星野 喬 Takashi HOSHINO

東京大学大学院情報理工学系研究科博士課程在学中．日本学術振興会特別研究員 DC. 2003 年東京大学工学部電子情報工学科卒業．2005 年東京大学大学院情報理工学系研究科修士課程修了．データベースシステムの研究に従事．本会，情報処理学会，ACM，IEEE 学生会員．

合田 和生 Kazuo GODA

東京大学生産技術研究所特任助手．2000 年東京大学工学部電気工学科卒業．2005 年東京大学大学院情報理工学系研究科博士課程単位取得満期退学．博士(情報理工学)．並列データベースシステム，ストレージシステムの研究に従事．本会，情報処理学会，ACM，USENIX 会員．

喜連川 優 Masaru KITSUREGAWA

1978 年東京大学工学部卒．1983 年同大大学院工学系研究科情報工学博士課程了．工学博士．同年同大生産技術研究所講師．現在，同教授．2003 より同所戦略情報融合国際研究センター長．データベース工学，並列処理，Ｗｅｂマイニングに関する研究に従事．現在，日本データベース学会理事，情報処理学会，電子情報通信学会 各フェロー．平成 11-14 年 ACM SIGMODJapan Chapter Chair, 平成 9,10 年電子情報通信学会データ工学研究専門委員会委員長．VLDB Trustee (97-02), IEEE ICDE, PAKDD, WAIM などステアリング委員, IEEE データ工学国際会議(ICDE2005) General Chair.