

P2P型ウェブコンテンツ共有における相関性を考慮したキャッシングシステムの実現

On Realizing a Caching System for P2P Web Contents Sharing System

中村 聡史^{*} 塚本 昌彦^{*}
西尾 章治郎^{*}

Satoshi NAKAMURA Masahiko TSUKAMOTO
Shojiro NISHIO

P2P ネットワークを利用したウェブコンテンツ共有では、コンテンツ取得における待ち時間短縮が重要な課題となる。このコンテンツ取得における問合せ時間を短縮するため、本稿ではコンテンツ間の相関性を考慮したコンテンツキャッシングシステムを実現し、評価実験によりシステムの有用性を明らかにする。

In a web contents sharing system on P2P network, it is important to shorten the waiting time for contents acquisition. In order to achieve this, in this paper, we realized a contents caching system which uses the correlation between contents to make the caching efficient, and make the usefulness of a system clear by some experiments.

1. はじめに

本研究グループで提案、実装しているウェブコンテンツ共有システム[1]では、WWW上で公開されている膨大なウェブコンテンツ[4]をユーザが収集し、Gnutella[3]のようなP2P型のネットワークで共有する。ウェブコンテンツを保存する際に、コンテンツ取得元のURLや、ページのタイトル、サイズ、更新日時、取得日時などのメタ情報をまとめて保存しておくことで、柔軟な検索を可能としている。共有されるコンテンツの種類は多岐にわたり、すでに公開されていない過去のコンテンツなども共有できるようにしている。また、ユーザがコンテンツを保存したときに、そのコンテンツがその後の検索要求における検索対象となるようにしている。特に、同じ興味をもつユーザ同士で共有ネットワークを構成することにより、効果的にコンテンツ検索および共有ができる。このウェブコンテンツ共有システムから構成されるP2Pネットワークをウェブコンテンツ共有ネットワークと呼んでいる。

ウェブコンテンツ共有ネットワークではコンテンツをバケツリレー式に転送するため、コンテンツ要求ピアからコンテンツ保持ピアまでのホップ数を低減し、コンテンツ取得における待ち時間を短縮することが重要な課題となる。本研究グループでは、コンテンツ取得までの待ち時間を短縮するため、コンテンツ間に従属関係や接続関係など相関性があるこ

とを利用した能動的キャッシング手法[2]を実現している。能動的キャッシング手法では、コンテンツ要求を受信したときにコンテンツの内容を解析し、連続してアクセスされる可能性の高いコンテンツを要求ピアに向けて転送する。このとき、コンテンツの転送範囲を、コンテンツ間の相関度と、コンテンツ要求ピアから保持ピアまでのホップ数を掛けた値で設定することにより、相関度に応じた転送を実現している。能動的キャッシング手法を利用することにより、コンテンツ取得までの待ち時間を短縮することはこれまでの研究でできているが、隣接するピア同士や、経路単位でキャッシュの内容を比較したときに、キャッシュの内容が似たようなものになるなど、過度の偏りが発生してしまい、非効率であるという問題があった。

そこで本研究では、キャッシュの偏りを改善し、コンテンツ要求における待ち時間を短縮する分散キャッシング手法を提案、実装する。分散キャッシング手法では、相互接続するピア同士がコンテンツの相関性などを考慮し、効果的にキャッシュを分散配置することで、コンテンツ取得要求におけるコンテンツ要求ピアからコンテンツ保持ピアまでのホップ数を低減する。また実験により、分散キャッシング手法の有用性を明らかにする。

2. 分散キャッシング手法

コンテンツ要求ピアから保持ピアまでのホップ数低減には、コンテンツを極力多くのピアに分散配置しておくことが重要となる。しかし、ウェブコンテンツ共有ネットワークでは、階層のないP2P型のネットワークを利用しているうえ、共有されるコンテンツの数が膨大すぎる場合には、コンテンツをトップダウン式に分散配置することはできない。また、ウェブコンテンツ共有ネットワークでは、コンテンツがバケツリレー式に転送されるため、コンテンツが転送される経路に同一のコンテンツがキャッシュされていく。そのため経路単位で比較した場合、キャッシュに無駄が多くなってしま

提案する分散キャッシング手法では、コンテンツ取得要求に必要なホップ数を極力増加させないようにするため、キャッシュ中のコンテンツに優先度を設定し、キャッシュ可能サイズの限界近くまでコンテンツを保存すると、優先度の低いコンテンツから順に破棄していく。

キャッシュ中に蓄えられたコンテンツの優先度は、下記の条件により与えられる。

- 最後にアクセスされてからの経過時間
- 他のコンテンツとの相関度
- 隣接ピアでの保存状態
- コンテンツのサイズ

転送されてきたコンテンツを受信したときや、コンテンツに対する検索問合せがあったとき、コンテンツに対する取得要求があったときは、そのコンテンツが必要とされている。必要とされているコンテンツは引き続きアクセスされる可能性が高い。そこで、最後にアクセスされてからの時間が長いコンテンツの優先度を高く、最後にアクセスされてから長い間アクセスされていないコンテンツは優先度を低く設定する。この優先度の設定により、あまりアクセスされていないコンテンツを効果的に破棄し、アクセスされやすいコンテンツをキャッシュ中に残すことが可能となる。

コンテンツ間に強い相関関係のあるウェブコンテンツ共有システムでは、コンテンツが連続的にアクセスされる可能性が高い。例えば、従属関係にあるコンテンツは、必ず連続

^{*} 正会員 情報通信研究機構 gon@nict.go.jp

^{*} 正会員 大阪大学大学院情報科学研究科 {tuka, nishio}@ist.osaka-u.ac.jp

してアクセスされるため、主となるコンテンツと従属するコンテンツは合わせて保持しておくことが好ましい。そこで、他のコンテンツとの相関度の高いコンテンツの優先度を高く設定することで、アクセスされやすいコンテンツをキャッシュ中に残すことが可能となる。

ウェブコンテンツ共有システムにおけるコンテンツ保存スタイルとしては、ユーザがデータとして明示的にローカルディスクに保存するタイプと、中継時にキャッシュとして一時的に保存するタイプがある。ユーザがデータとしてコンテンツを保存している場合は、ユーザが削除処理を行わない限りコンテンツが消えてしまうことはないが、キャッシュに保存されているコンテンツは、キャッシュの状態に応じてシステムに自動的に削除されてしまう。ここで、自ピアにおいてキャッシュに保存されているコンテンツが、隣接するピアでデータまたはキャッシュとして保存されている場合は、キャッシュからそのコンテンツを削除しても、コンテンツ取得要求におけるホップ数は最悪で1しか増加しない。そこで、自ピアでキャッシュされているコンテンツが、隣接するピアで保存されている場合は優先度を低く、隣接するピアで保存されていない場合は優先度を高く設定することにより、コンテンツが分散配置され、コンテンツ取得要求における待ち時間を短縮できると考えられる。このとき、コンテンツを保持している隣接ピアの数が多ければ多いほど優先度は低く、少なれば少ないほど優先度は高く設定することが好ましい。また、コンテンツがデータとして保存されているか、キャッシュとして保存されているかによって、該当コンテンツが残留する確率は異なる。そこで、データとして保存されていれば優先度を低く、キャッシュに保存されている場合は優先度を高く設定することが有効である。一方、隣接ピアでの保存状態のチェックは、負荷の増大を招く可能性がある。

ウェブコンテンツ共有システムで共有されるコンテンツは、WWW上で公開されているコンテンツを想定しているため、コンテンツの種類はテキストから画像、音声、映像など多岐にわたり、コンテンツのサイズも、1KBytes程度のテキストから10MBytes程度の映像までさまざまである。ここで、キャッシュ可能サイズが100MBytesとしたとき、1Kbytesのコンテンツはキャッシュに10万個保持できるが、10MBytesのコンテンツはキャッシュに10個しか保持することができない。全てのコンテンツが同じ確率でアクセスされると、ファイルサイズの大きなコンテンツよりは、ファイルサイズの小さいコンテンツを多数キャッシュに保持する方が、そのピアにおけるキャッシュヒット率は高くなる。つまり、単純にキャッシュヒット率を向上させるためには、ファイルサイズの小さなコンテンツをキャッシュに優先的に残す方が良く考えられる。ただし、すべてのピアがファイルサイズの小さなコンテンツをキャッシュに優先的に残すような戦略をとると、ファイルサイズの大きなコンテンツはどこのピアでもキャッシュされにくくなる。また、このファイルサイズの大きなコンテンツは、ファイルサイズの小さなコンテンツに比べ、ネットワークを介したコンテンツ転送にも余計に時間がかかってしまう。結果として、ネットワーク全体で見たときのパフォーマンスが低下する可能性がある。そこで、ネットワークの帯域が狭いときにはファイルサイズの大きなコンテンツを優先的に保持し、キャッシュ可能サイズに余裕がないときにはファイルサイズの小さなコンテンツを優先的に保持するなどして、各ピアでファイルサイズに関するコンテンツの優先度の付け方を変更する機構が必要となると考えられる。

以上のように、キャッシュ中におけるコンテンツの優先度は、最後にアクセスされてからの経過時間、他コンテンツとの相関度、隣接ピアでの保存状態、コンテンツのサイズなどにより決定される。

3. 実装

実装する分散キャッシング手法では、キャッシュ可能サイズの限界に達すると、キャッシュ中に含まれるコンテンツの優先度の低いコンテンツから順に削除する。ウェブコンテンツ共有システムで共有されるコンテンツの量が膨大な場合が考えられるうえ、コンテンツのサイズが他のコンテンツ共有システムで共有されるコンテンツのサイズに比べ非常に小さく（平均10KBytes程度）、キャッシュ可能サイズを100MBytes程度に制限しても、キャッシュされるコンテンツの数は1万に達する。キャッシュシステムは、優先度の低いコンテンツを即座に求め、該当するコンテンツをキャッシュから迅速に削除する仕組みが必要となるが、2章で述べたすべての条件を考慮し、随時優先度の低いコンテンツを計算するのは困難である。

そこで本研究では、コンテンツの優先度について、最後にアクセスされてからの経過時間と、他コンテンツとの相関度、隣接ピアでの保存状態について考慮する。プロトタイプシステムでは、キャッシュ中のすべてのコンテンツを、最後にアクセスされてからの経過時間をベースとしたリスト形式（キャッシュリスト）で保持する。キャッシュリストには、コンテンツ自体の情報（URLや最終更新日、サイズなど）のほかに、タイムスタンプ、繰り返し回数という値を用意する。キャッシュリストには先頭から追加し、末尾から削除するものとする。コンテンツ自体は、URLと日時をベースとしたキャッシュフォルダに保存し、保存時やアクセス時に、ファイルシステムにおいてコンテンツのファイルの最終更新日を設定する。また、キャッシュ可能サイズとキャッシュ使用サイズを保持する。なお、タイムスタンプというのは該当するコンテンツがアクセス（受信、送信、保存など）された時刻を意味する。さらに、繰り返し回数というのは、コンテンツのキャッシュ中における優先度をベースにした値で、コンテンツがキャッシュリストの末尾に達し、削除対象となったときに、再度キャッシュの先頭への追加を許可するかどうかを判断するための値である。この繰り返し回数と経過時間をベースとしたリスト形式のキャッシュリストを利用することで、相関性などを考慮した優先度付きの分散キャッシングシステムを実現する。

隣接ピアでの保持状態については、隣接ピアにその都度問合せを行い、優先度を計算するのは非現実的である。コンテンツが転送される場合は、コンテンツが転送されたホップ数（Hops）、あて先までのホップ数（TTL）が指定されてコンテンツが送信される。コンテンツ中継時に、Hopsが1の場合、コンテンツは隣接ピアでデータまたはキャッシュとして保存されていたと考えられる。ここで、Hopsが奇数ホップ目のピアではコンテンツを積極的に保持せず、偶数ホップ目ではコンテンツを積極的に保持するという戦略をすべてのピアがとると、互いにキャッシュ問い合わせをすることなく、キャッシュの分散配置が可能となる。一方、TTLについても、TTLが1の時は隣接ピアであるコンテンツ要求ピアでコンテンツが保持されると予想されるため、Hopsと同様の戦略を取ることで、キャッシュの分散配置が可能となる。以上の仕組みにより、隣接ピアでのコンテンツの保存状態を

予測しながらキャッシュを最適化する。なお、コンテンツ取得要求や能動的キャッシングにおいてコンテンツが転送される際、等しい相関度をもつコンテンツは、コンテンツ保持ピアから同じホップ数だけ転送される。結果として、従属するコンテンツなど相関性のあるコンテンツは同じピアに保持されやすくなるため、相関性を考慮した分散キャッシュが実現できる。

Hops と *TTL* については、コンテンツ保持ピアの方が、コンテンツ要求ピアに比べ、コンテンツをデータとして保存している可能性が高いと考えられるため、*Hops* を優先してコンテンツに繰り返し回数を設定する。一方、コンテンツ保持ピアからコンテンツ要求ピアまでのホップ数が大きいとき、中間のピアでコンテンツを長く保持することが好ましいと考えられる。*Hops* と *TTL* の和により、中継ピアにおいてコンテンツ保持ピアからコンテンツ要求ピアまでのホップ数が求まるため、*Hops* と *TTL* の和が大きく、ピアが両者の中間地点に存在するとき、コンテンツを長く残留させるようにする。本研究では、*Hops* や *TTL* が 0 の場合コンテンツの繰り返し回数を 2、1 のときは繰り返し回数を 0 と設定する。また、中間地点などにコンテンツを長く残留させるため、*Hops* と *TTL* の和が 8 以上で、*Hops* が 4 の倍数かつ *TTL* が 4 以上のとき、繰り返し回数を 2 に設定する。上記の条件を満たさず、*Hops* が偶数の時には繰り返し回数を 1 に、奇数のときは繰り返し回数を 0 と設定する。

コンテンツを取得または送信すると、同一のコンテンツがキャッシュ中に含まれているかどうかを判定する。同一のコンテンツがキャッシュに含まれていない場合は、コンテンツをキャッシュフォルダに格納した後、キャッシュリストの先頭にコンテンツの情報を格納する。このとき、キャッシュリストのタイムスタンプとキャッシュ中のファイルの最終更新日を同一のものに設定する。また、キャッシュの繰り返し回数も *Hops* や *TTL* をもとにして同様にセットする。さらに、キャッシュ使用サイズに追加したコンテンツのサイズを加算する。すでに同一のコンテンツがキャッシュに含まれている場合は、コンテンツの最終更新日を現在時刻に設定し、キャッシュリストの先頭にコンテンツの情報を格納する。コンテンツに対する取得要求や検索問合せが発生したときには、キャッシュ中のコンテンツの最終更新日を変更し、キャッシュリストの先頭にそのコンテンツの情報を追加する。

一方、コンテンツ追加時にキャッシュ可能サイズとキャッシュ使用サイズを比較し、キャッシュを限界近くまで使用していないかを調べる。キャッシュ可能サイズにキャッシュ使用サイズが近づくと、新しいコンテンツを保存するための余裕がなくなるため、キャッシュリストから削除するコンテンツを決定し、キャッシュフォルダからコンテンツを削除する。なお、キャッシュ使用サイズがキャッシュ可能サイズを超えないようにするため、キャッシュ使用サイズがキャッシュ可能サイズの 95% に達したとき、キャッシュ削除試行を実施する。この割合をキャッシュ削除開始割合と呼ぶ。この割合はユーザごとに設定できる。

キャッシュを削除する場合は、まず始めにキャッシュリストの末尾にあるコンテンツ情報をリストから取り出し、コンテンツのタイムスタンプと、そのコンテンツに対応するキャッシュフォルダ中のファイルの最終更新日を比較する。タイムスタンプとファイルの最終更新日が一致しない時はこのキャッシュ情報を破棄する。一方、コンテンツの最終更新日とタイムスタンプが一致する場合は、コンテンツ情報に含まれる繰り返し回数をチェックし、繰り返し回数が 1 以上の場

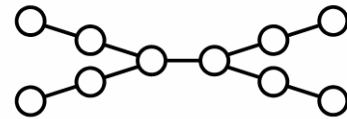


図1 実験で使用したネットワークトポロジ
Fig.1 The network topology for experiment.

合は、繰り返し回数を 1 つ減らしてキャッシュリストの先頭にそのキャッシュ情報を追加する。繰り返し回数が 0 の場合は、キャッシュフォルダからコンテンツを削除し、キャッシュ使用サイズをコンテンツサイズ分減らす。また、コンテンツのキャッシュ情報も破棄する。この試行を、キャッシュ使用サイズがキャッシュ可能サイズの 90% を下回るまで繰り返す。この割合をキャッシュ削除停止割合と呼ぶ。この割合についてもユーザごとに設定できるようにしている。

本実装では、キャッシュリスト中に同一コンテンツに関する複数のキャッシュ情報が登録されるが、前に登録されたキャッシュ情報のタイムスタンプと、コンテンツの最終更新日が異なるため、そのキャッシュ情報を処理するときは無視する。このような実装により、キャッシュリストの無駄な走査が不要となり、ピアにおける負荷を軽減できると考えられる。

4. 実測実験

プロトタイプシステムの有用性を示すため、コンテンツを中継する際にコンテンツをキャッシュに保存し、一定期間経過後キャッシュから削除する従来手法と、能動的キャッシング手法、分散キャッシング手法、能動的キャッシング手法と分散キャッシング手法の併用について比較を行った。それぞれの手法について、実機を用いたネットワークを利用して、キャッシュ可能サイズを変化させながらコンテンツ取得における待ち時間がどのように変化するかを計測した。

実験では、10 台のピアを用意し、図 1 のようなトポロジの P2P ネットワークを構成する。また、コンテンツとして 4,000 個のウェブページ（ファイル数は約 28,000、総サイズは約 480MBytes）を用意し、10 台の各ピアに他と重複しない 400 個のウェブページと、他と重複する適当に選ばれた 100 個のウェブページをデータとして保存しておく。各ピアはコンテンツ問合せに利用するため、すべてのコンテンツの URL を保持しておき、各ピアは、30 秒ごとに問合せを行うタイミング（0~10 秒の間で決定）と、取得するコンテンツをリストからランダムに決定し、コンテンツの取得を行う。なお、取得対象となるコンテンツを、データとして保持している場合は、改めて別のコンテンツを選択する。

コンテンツ取得の前には、実際に URL を利用したコンテンツの検索問合せを行い、最初に返事が返ってきたピアに対してコンテンツ取得要求を送信する。本実験では、コンテンツ取得要求を送信してから、インラインオブジェクトも含めたウェブページを完全に受信できるまでの時間を、各ピアにおいて計測する。この試行を 1 時間にわたり実施する。

時間の経過によるコンテンツ取得の待ち時間の変化は、図 2 と図 3 の通りである。図の横軸は時間で、縦軸はすべてのピアにおけるコンテンツ取得待ち時間の平均を表している。図 2、図 3 ではそれぞれ、キャッシュ可能サイズを 10MBytes、20MBytes に制限している。図より明らかなように、コンテンツが随時キャッシュされていくことで、コンテンツ取得における待ち時間が短縮されている。また、キャッシュ可能サイズに余裕があるときに比べ、キャッシュ可能サイズに余裕

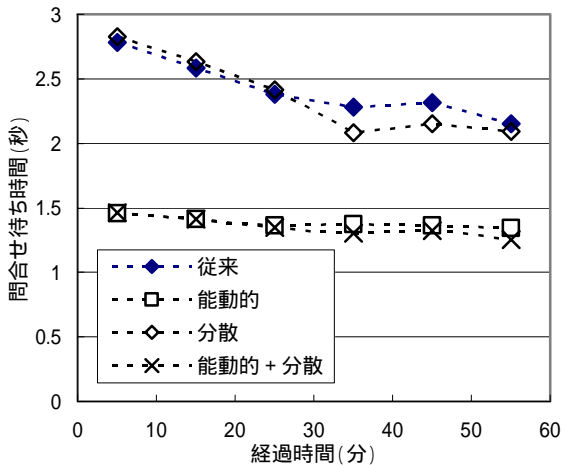


図2 実験結果 (キャッシュ可能サイズ 10MB)
Fig.2 The results of experiment. (10MBytes).

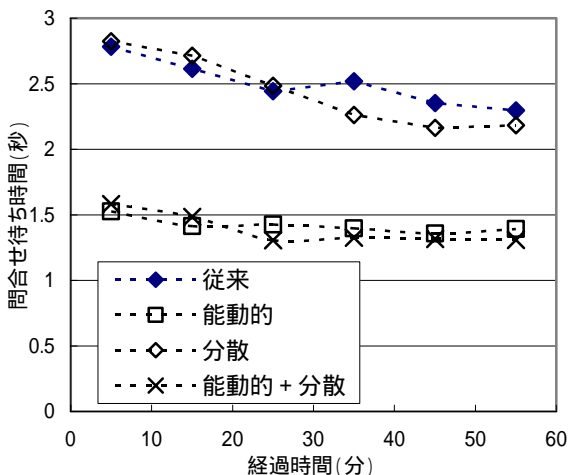


図3 実験結果 (キャッシュ可能サイズ 20MB)
Fig.3 The results of experiment. (20MBytes).

がないときは、良い結果が出ていないことがわかる。

各手法を比較すると、従来手法に比べ、能動的キャッシング手法を利用するとコンテンツ取得における待ち時間が大幅に短縮できていることがわかる。これは、ホップ数の低減が効率よく働いているためと考えられる。一方、分散キャッシング手法については、従来手法に比べ、良い結果が得られていることがわかる。また、従来手法や能動的キャッシング手法がキャッシュサイズの変化により影響を受けているのに比べ、分散キャッシング手法ではキャッシュサイズの変化にあまり影響を受けていないことがわかる。

今回の実験では、ウェブコンテンツ共有ネットワークにおけるコンテンツ取得の待ち時間短縮に、コンテンツの能動的キャッシングが特に有効であることがわかったが、分散キャッシング手法の有用性については明らかにすることができなかった。これは、分散キャッシング手法は、コンテンツ取得要求ピアからコンテンツ保持ピアまでのホップ数が長く、長時間での運用により効力を発揮するのに対し、今回実施した実験では、十分なネットワークサイズや時間を確保できなかったことが理由として考えられる。また、あらかじめ用意したコンテンツの量が多くないことや、キャッシュ可能サイ

ズも比較的小さく設定されていたことも分散キャッシング手法にとって不利だったと考えられる。今後の実験では、長期的な実験を行うことで、それぞれの特徴について明らかにする予定である。

5. まとめと今後の課題

本稿では、ウェブコンテンツ共有システムを利用したコンテンツ共有におけるコンテンツ取得の待ち時間を短縮するため、コンテンツ間の相関性を考慮した分散キャッシング手法を提案し、キャッシング手法を用いたシステムを実現した。また、実測実験を実施することにより、実際にウェブコンテンツ共有ネットワークにおけるコンテンツ取得までの時間を短縮できていることを確認した。

今回行った実測実験では、小規模なネットワークについてしか調査を行うことができなかった。今後は、実測実験の結果を元にモデルを作成し、大規模な P2P 環境におけるシミュレーションを実施する予定である。また、大規模な P2P 環境における実測実験も行い、計算負荷やネットワーク負荷などについて調査する予定である。一方、コンテンツの相関度を詳細に計算する方法や、最適な P2P ネットワークを形成する方法についても考えていく予定である。

【謝辞】

本研究は、文部科学省 21 世紀 COE プログラム (研究拠点形成費補助) の研究助成によるものである。ここに記して謝意を表す。

【文献】

- [1] 中村聡史, 塚本昌彦, 西尾章治郎: “コンテンツ流通制御を考慮したウェブコンテンツ共有システムの実現,” 情報処理学会論文誌, Vol.45, No.1 (2004).
- [2] 中村聡史, 塚本昌彦, 西尾章治郎: “ウェブ共有環境におけるコンテンツ間の相関性を考慮したキャッシングについて,” 情報処理学会研究報告, Vol.2004, No.3, pp.66 - 72 (2004).
- [3] Gnutella, <http://www.gnutella.com/>.
- [4] 総務省, “平成 15 年版情報通信白書,” (2003).

中村 聡史 Satoshi NAKAMURA

情報通信研究機構所属。2004 年大阪大学大学院工学研究科情報システム工学専攻博士後期課程修了。日本データベース学会、ヒューマンインタフェース学会の会員。

塚本 昌彦 Masahiko TSUKAMOTO

大阪大学大学院情報科学研究科マルチメディア工学専攻助教授。1994 年京都大学院工学研究科博士後期課程修了。工学博士。ウェアラブル、ユビキタスコンピューティングの研究に従事。ACM, IEEE など 8 学会の会員。

西尾 章治郎 Shojiro NISHIO

大阪大学大学院情報科学研究科マルチメディア工学専攻教授。1980 年京都大学院工学研究科博士後期課程修了。工学博士。2000 年より大阪大学サイバーメディアセンター長、2003 年より大阪大学大学院情報科学研究科長を併任。データベース、知識ベース、分散システムの研究に従事。現在、Data & Knowledge Engineering, Data Mining and Knowledge Discovery などの論文誌編集委員。情報処理学会フェローを含め、ACM, IEEE など 9 学会の会員。