iSCSIストレージアクセスのトレースシステム

Trace System of iSCSI Storage Access

山口 実靖♥ 小口 正人♦
喜連川 優♥

Saneyasu YAMAGUCHI Masato OGUCHI
Masaru KITSUREGAWA

本稿では iSCSI を用いた IP-SAN のアクセストレースシステムを提案し，その適用例を紹介し有効性を示す．FC-SAN の欠点を補う SAN として TCP/IP と Ethernet を用いた IP-SAN が期待を集めているが，IP-SAN は性能が FC-SAN よりも劣るとの欠点も指摘されており性能の向上が重要であると考えられる．IP-SAN は，多段のプロトコルスタックで構成されており，さらにサーバ計算機とストレージ機器が協調して動作するという非常に複雑な構造のため性能劣化原因の発見が困難となっている．そこで本稿では，これらの多段プロトコルスタックの全層の振る舞いを観察可能であり，サーバ計算機とストレージ機器の双方を統合的に観察可能な IP-SAN トレースシステムを提案し，その実装の紹介を行う．そしてこれを実際に高遅延環境における並列 iSCSI アクセスに適用したところ，性能制限原因の発見および発見された問題の解決により並列 iSCSI アクセスの性能を向上させられることが確認され，提案システムの有効性が示された．

In this paper, we propose an IP-SAN access trace system and demonstrate that performance can be improved by using the system. IP-SAN and iSCSI are expected to remedy problems of FC-based SAN. In IP-SAN systems using iSCSI, servers and storages work cooperatively by communicating with each other via TCP/IP, thus integrated analysis of servers and storages can be considered important. We explain our integrated trace system and show that the system can point out the cause of performance degradation.

1. はじめに

計算機システムの運用の大きな問題点の１個として，ストレージの管理費用の大きさが指摘されている．ストレージの運用には定期的なバックアップ等の作業が必要となり大きな管理費用が必要となる．この問題の解決策として SAN (Storage Area Network) の導入が提案された．SAN はストレージ専用の高速ネットワークであり，１ヶ所で管理されているストレージに対して各計算機から SAN を用いて接続しストレージを使用する．ストレージを計算機の周辺機器として個別に管理するのではなく，SAN を用いてストレージを１箇所に集約することによりその管理費用は大幅に削減されることが期待される．この効果は高く評価をされており，現在多くの企業において SAN が導入されている．しかし現在の FC (Fibre Channel) を用いる FC-SAN は普及に伴い，以下のような欠点も明らかとなってきた．すなわち，(1)FC の管理技術を持つ技術者が少ない，(2)FC の導入費用は高い，(3)FC は接続距離に限界がある，(4)FC は相互接続性が必ずしも高くない，などの問題点も指摘されるようになり，これらの問題点を解決する SAN として TCP/IP と Ethernet を用いて構築する IP-SAN や，その標準的データ転送プロトコルである iSCSI [1]に期待が集まるようになってきている．iSCSI を用いた IP-SAN では SCSI プロトコルを TCP/IP の中にカプセル化し Ethernet を用いて SCSI アクセスを転送する．よって，(1)TCP/IP や Ethernet の管理技術を持つ技術者が多い，(2)導入費用が低い，(3)接続距離に限界がない，(4)相互接続性が高い，などの利点が期待されている．逆に IP-SAN の欠点としては，FC-SAN と比べて性能が劣ることが指摘されており，この問題の解決が最も重要であると言える．特にネットワーク遅延による性能の劣化の問題が指摘されており[2]，本研究では高遅延環境における性能について考察をする．

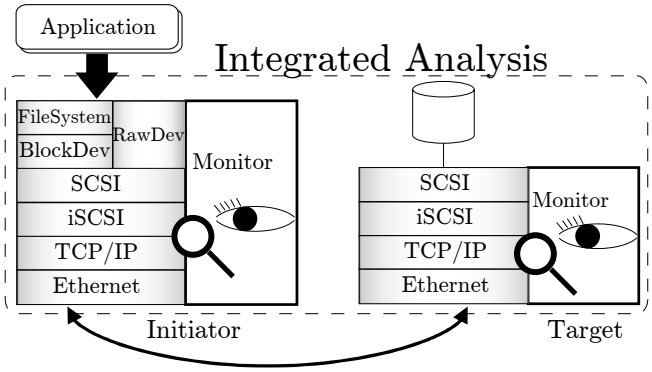

図１ iSCSI プロトコルスタックと解析システム

Fig.1 iSCSI Protocol Stack and Analysis System

iSCSI を用いた IP-SAN では図１の様にプロトコルスタックが SCSI over iSCSI over TCP/IP over Ethernet という複雑な階層構造をとるが，End-to-End のストレージアクセスはこれら全層を通過して行われるため，IP-SAN の性能を向上させるためにはこれら全層の振る舞いの把握が重要であると考えられる．さらに，サーバ計算機とストレージ機器が独立したシステムとして個別に動作しているが iSCSI アクセスは双方のプロトコルスタックを合成して構築されているためこれらの一方のみの解析ではシステム全体の振る舞いを把握することは困難であり，双方を統合的に解析するシステムの実現が重要であると考えられる．本稿では iSCSI ストレージアクセスを構成するこれら全層の解析が可能であり，サーバ計算機，ストレージ機器のトレース結果を統合的に処理することを可能とするトレースシステムを提案する．そしてそれを高遅延環境における並列 iSCSI アクセスに対し実際に適用し性能劣化原因の発見と性能向上が可能であることを示す．

本稿は，以下のように構成されている．第２章において本稿で提案する IP-SAN のトレースシステムの紹介を行い，第３章において提案したトレースシステムを実際に並列 iSCSI

♥ 正会員 東京大学生産技術研究所
{sane,kitsure}@iis.u-tokyo.ac.jp
♦ 正会員 お茶の水女子大学理学部情報科学科
oguchi@computer.org

アクセスに対し適用し当システムの有効性を示す．第 4 章において関連する研究を紹介し，最後に，第 5 章において本稿のまとめを述べる．

2. IP-SAN トレースシステム

本稿で提案する“IP-SAN トレースシステム”は，図 1 の様な構造をしている．オープンソース OS 実装(Linux 2.4.18)とオープンソース iSCSI 実装(ニューハンプシャー大学の InterOperabilityLab[3]が配布する iSCSI 実装 ver.1.5.02)を用いて IP-SAN 環境を構築し，各層にその振る舞いをトレースできるモニタコードを適用する．そして，モニタされた各層の振る舞いを統合的に解析し，アプリケーションによる I/O 要求の発行から HDD デバイスまでの全振る舞いの把握を可能とする．図 2 に，当システムを用いて iSCSI シーケンシャルアクセスのトレース結果を可視化した例を示す．同図の縦軸は iSCSI アクセスのプロトコルスタックの遷移を表している．すなわち，上から順に，(1)アプリケーションによるシステムコールの発行，(2)raw デバイス層，(3)SCSI 層，(4)iSCSI 層，(5)TCP/IP 層，(6)Ethernet によるパケットの転送，(7)TCP/IP 層，(8)iSCSI 層，(9)SCSI 層，(10)HDD デバイスアクセス，を表している．(1)～(5)が，サーバ計算機内における処理であり，(7)～(10)がストレージ機器における処理である．本トレース例では，ファイルシステムを用いずに raw デバイスを用いた．また使用した iSCSI Target は“ファイルモード”で動作させたため最下位層の HDD デバイスアクセスは実際はファイルアクセスの実行のトレースとなっている．横軸が時間の経過を表しており，iSCSI ストレージアクセスの各処理における消費時間等を視覚的に確認することが可能となる．また，大きなブロックサイズのシステムコールが各層で細分化されている様子や，待ち状態にある処理の把握なども可能となる．同図の例では 2MB のシステムコールが発行されており，これを raw デバイスが 512KB ごとの 4 要求に分割し，4 要求が完了した時点で上位層にシステムコールの完了を通知していることや，raw デバイスから発行された 512KB の要求が 32KB の SCSI 命令に分割されていること，細分化された要求を用いてネットワークに処理要求が送出されている様などを観察することが可能である．

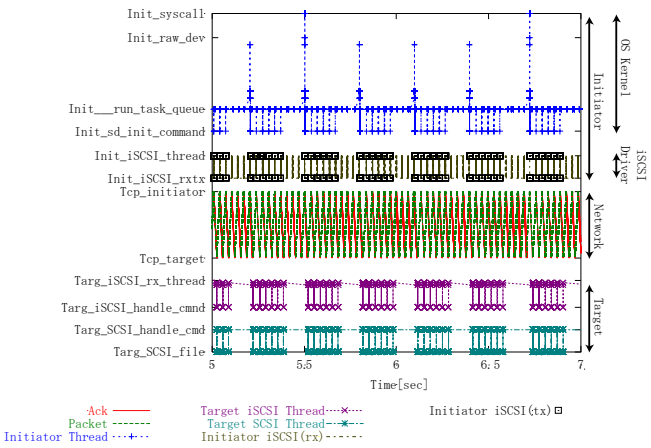

図 2　並列 iSCSI アクセスのトレース：A

Fig.2　Visualized Trace of Parallel iSCSI Access: A

3. 提案システムの評価

本章では提案解析システムを実際に高遅延環境下における並列ショートブロック iSCSI リードアクセスに適用し，その有効性を示す．

3.1 並列アクセス実験

サーバ計算機-ストレージ機器間の片道遅延時間が 16ms の環境下においてベンチマークプログラムを複数プロセス同時に動作させ，全プロセスの合計性能を計測した．各ベンチマークは iSCSI 接続の raw デバイスに対し 512 バイトのシステムコール read()をシーケンシャルに 2048 回ずつ発行する．iSCSI ターゲットは“ファイルモード”で動作させ，ファイル内容が実メモリ上にキャッシュされている状態で計測を行った．よって I/O 要求は必ずストレージ機器の SCSI 層まで到達し，メモリ上のキャッシュをヒットすることになる．上記の実験を行い，図 3 の“can_queue=2(default)”の結果を得た．

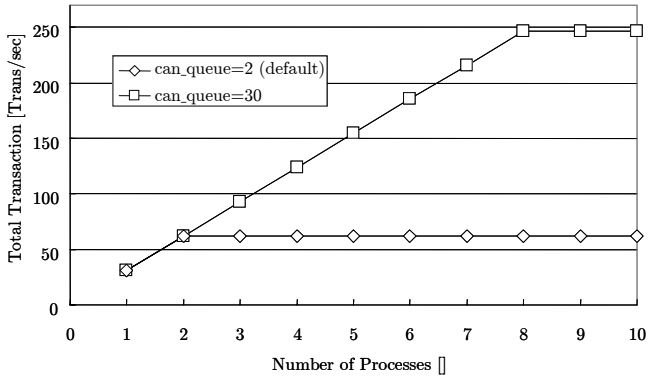

図 3　並列 I/O の合計性能列

Fig.3　Total Performance of Parallel I/O

同図横軸は並列に動作させたベンチマークプロセスの数である．縦軸は合計性能を表し，単位時間における全プロセスの合計トランザクション数である(トランザクション数は“512 バイトのシステムコール回数”)．同結果より，プロセス数の増加に伴う合計性能の向上は，並列度 2 において飽和となり，iSCSI プロトコルスタックのいずれかの層において並列度が 2 に制限されていると予想される．

3.2 トレースシステムによる解析

本節において前節の実験の 3 プロセス並列アクセスのトレース解析を示す．

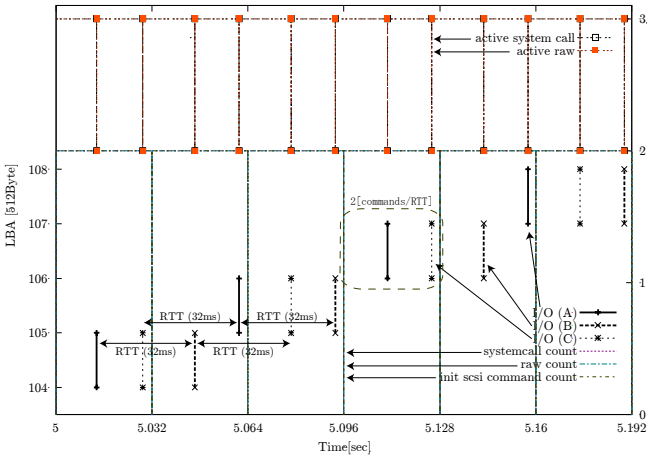

図 4　並列 iSCSI アクセスのトレース：A

Fig.4　Trace of Parallel iSCSI Access: A

まず，①イニシエータの SCSI 層において発行された

SCSI Read 命令のアドレス，②各往復時間(32ms)内に発行されたシステムコール数, raw デバイスからの要求数, SCSI 層における SCSI 命令の数，③同時に処理中(すでに要求されたが終了していない)にあるシステムコールの数，raw デバイスにおける要求の数, の時間変化を可視化し図 4 を得た．①は，3 プロセスそれぞれ I/O(A)，I/O(B)，I/O(C) で記されており 1 往復時間内に 2 個の SCSI 命令しか発行されていないこと，ある命令発行の 1 往復時間後にその処理が終了し次の命令が発行されていることなどが確認できる．②は“system call count”，“raw count”，“init SCSI command count”で表されておりそれぞれ各往復時間内において発行されたシステムコール数, raw デバイスからの要求数, SCSI 命令の発行数である．これらからも 1 往復時間内において各処理は 2 個ずつしか実行されていないことが確認できる．それに対し，③処理中のシステムコール，raw デバイスによる要求の数(同図における“active system call”，“active raw”)は常に 3 となっている．図より処理途中のシステムコール，raw デバイス要求数は多くの時刻において 3 であり，処理が終了した瞬間(同図の例においては 5.014 秒，5.029 秒など)に 2 に減少しその直後に次の要求が発行され処理途中要求数は再度 3 に上昇しているのが確認できる(ただし，“active system call”と“active raw”はほぼ同時刻に変化するため同図内では両線は重なって表示されている)．以上より，常に 3 個のシステムコールが待ち状態にあるが，1 往復時間で 2 要求ずつしか処理されていないことがトレースの解析により確認された．

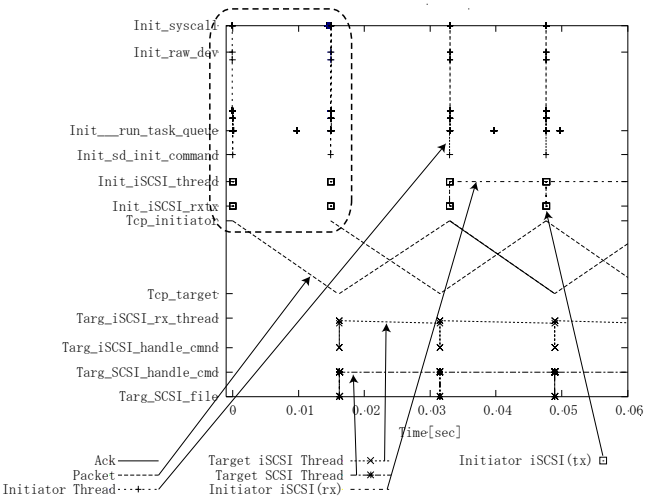

図 5　並列 iSCSI アクセスのトレース：A

Fig.5　Trace of Parallel iSCSI Access: A

次に，並列度制限に関する巨視的な解析結果を示す．図 5 が, 上記実験のプロセス数 3 における iSCSI ストレージアクセの可視化結果である．同図より各往復時間内にサーバ計算機からストレージ機器に対して 2 個の I/O 要求しか送出されていないことが確認でき，並列数 2 の制限はサーバ計算機側に存在することが分かる．また，図 5 の破線部を拡大し表示すると，図 6 の左上の様になる．図 6 左上よりシステムコールはストレージ機器からの応答を待つことなしに 1 往復時間(32ms)内に 3 個発行されていることや, ベンチマーク“I/O(A) ”の要求は時刻 0.000 秒に発行され raw デバイス層，SCSI 層，iSCSI 層を経由し，TCP/IP 層に至りストレージ機器に送出されていることが確認できる．図 6 左上の破線部を拡大することにより，同図右下が得られる．図 6 右下よりベンチマーク“I/O(C) ”のシステムコールは時刻 0.01485 秒に発行され，同様にネットワークに送出されていることが確認できる．これに対し，ベンチマーク“I/O(B) ”では，システムコールが時刻 0.01494 秒に発行され直後に raw デバイス層を通過しているが, SCSI 層の SCSI 命令の発行に至っておらず，SCSI 命令の同時発行上限が 2 となっていることが確認できる．

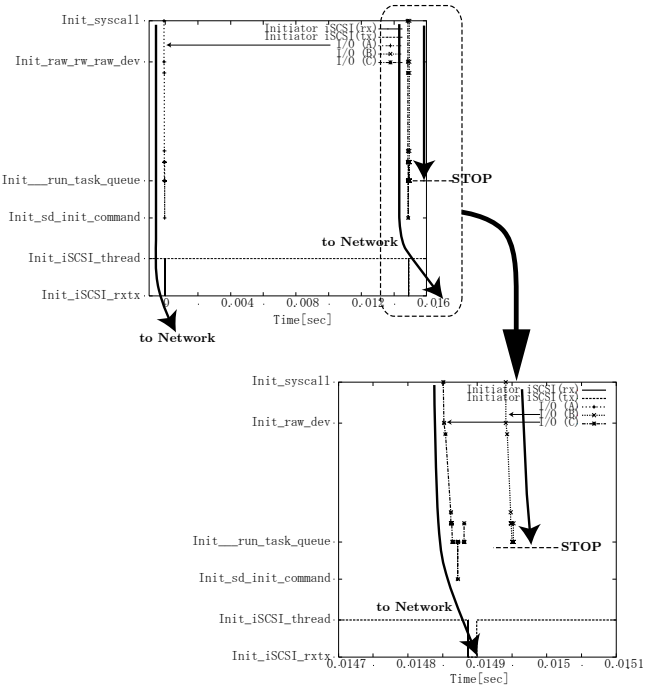

図 6　並列 iSCSI アクセスのトレース：B

Fig.6　Visualized Trace of Parallel iSCSI Access: B

“drivers/scsi/scsi_lib.c”

```
851 void scsi_request_fn(request_queue_t * q)
852 {
872  while (1 == 1) {
895   if ((SHpnt->can_queue > 0
          && (atomic_read(&SHpnt->host_busy) >= SHpnt->can_queue))
896       || (SHpnt->host_blocked)
897       || (SHpnt->host_self_blocked)) {
911    break;
912   } else {
914    atomic_inc(&SHpnt->host_busy);
916   }
1015  if (SCpnt->request.cmd != SPECIAL) {
1046   if (!STpnt->init_command(SCpnt)) {
1064   }
1065  }
1102 }
1103 }
```

Issuing SCSI command

-----→ host_busy>=can_queue
——→ host_busy< can_queue

図 7　Linux SCSI 層のトレース

Fig.7　Trace of Linux SCSI layer: “drivers/scsi/scsi_lib.c”

次に微視的な解析結果を示す．SCSI 命令が発行される最初の 2 要求と，発行されない 3 個目の要求のトレースの分岐点は Linux SCSI 層実装における図 7 の部分である．同実装は現在のアクティブな命令数“host_busy”と下位層(iSCSI 層)が同時に受け付け可能である命令数“can_queue”の比較部である．最初の 2 要求(I/O(A),(C))では host_busy がそれぞれ 0，1 であり，can_queue が 2 である．よって，“host_busy<can_queue”に示される処理(914 行目において host_busy をインクリメントし 1046 行目において SCSI 命

令を発行する)が記録された．3 個目の要求(I/O(B))では host_busy が 2 であり，“host_busy=>can_queue”に示される処理(SCSI 命令を発行しない)が記録された．よって，iSCSI 実装の can_queue の値が 2 であることが合計性能制限の理由であると予想される．そこで，can_queue 値を 30 に設定し(初期値は 2 である)性能を測定し，図 3 の“can_queue=30”を得た．同測定では合計性能は並列数 8 までほぼ線形に上昇しており，同例においてはトレースシステムを適用し合計性能の限界を約4倍に向上させることが可能であった．

4. 関連研究

iSCSIを用いたIP-SANの性能の評価に関する既存の研究として文献[2,4,5,6]があげられる．文献[2]は早期に SCSI over IP の性能評価を行った開拓的な研究である．独自の SCSI over IP 実装を用い遅延のある環境におけるシーケンシャル/ランダムアクセスなどの基本性能の評価や，アプリケーション性能の評価を行っている．また，ファイルシステム等が性能に与える影響などについても考察を行っている．Sarkar らは文献[4]において低遅延環境における iSCSI アクセスの性能についての評価を行っている．特に iSCSI 使用時における CPU 使用率に注目し考察行っている．文献[5]は，IP 接続ストレージのアクセス手法として iSCSI と NFS の比較を行っている．基本性能や応用性能の測定を行い iSCSI, NFS 双方の性能や CPU 使用率の評価を行っている．以上の研究は各種状況における iSCSI 性能の評価を行ったものであり，iSCSI の性能を知る上で有用な研究であると言えるがシステムの外部から負荷を与え性能を評価したものでありシステムの振る舞いについて考察を行ったものではない．よって，性能が高い理由や低い理由，性能の向上方法について十分な考察を行ったものではなく，性能向上方法の提供を目指す本研究とは目的が異なる．藤田らの文献[6]は，iSCSI ターゲットの内部の実装手法も考慮して iSCSI 性能の評価を行い，OS のカーネルに変更を加える手法や低レベルインターフェイスを使用する実装手法が性能において優れていることを指摘している．ターゲット実装に着目し詳細な考察を行っている点において，システム全体の考察を目指す我々の研究と目的が同じでは無いが，ターゲットシステム実装の詳細な考察を行った既存の研究として価値が高いと思われる．

5. まとめと今後の課題

本稿では，サーバ計算機とストレージ機器の分散協調システムとして動作する IP-SAN システムのアクセストレースシステムを提案し，その有効性の検証を行った．その結果，提案システムは多段プロトコルスタックの中から性能劣化原因を的確に発見することが可能であり，その回避により多並列アクセス時に 4 倍の性能向上が実現され，提案手法が IP-SAN システムの性能向上の実現に有効な手法であることが確認された．

本稿では解析対象としてファイルシステムやブロックデバイスを用いない例を選択し，ストレージ機器の HDD デバイスとしてもファイルモードを採用した．しかし，ファイルシステムやブロックデバイスのキャッシュヒットによるネットワークより上位層における要求の終了など性能に大きな影響を与える振る舞いの把握も重要であると考えられる．また，ショートブロックアクセス時には HDD デバイスアクセス時間の考察も重要であると考えられる．今後はより実応用に近いシステムへのトレース解析の適用を考え，ファイルシステムや実 HDD デバイスを用いたシステムの解析を行っていく．そして，トレースシステムの適用に起因するオーバーヘッドなどについても考察していく．

[文献]

[1] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, E. Zeidner: “Internet Small Computer Systems Interface (iSCSI)”, IETF RFC 3720 http://www.ietf.org/rfc/rfc3720.txt (2004).

[2] Wee Teck Ng, Bruce Hillyer, Elizabeth Shriver, Eran Gabber and Banu Ozden: “Obtaining High Performance for Storage Outsourcing”, Proceedings of FAST 2002 pp. 145-158 (2002).

[3] University of New Hampshire InterOperability Laboratory iSCSI Consortium, http://www.iol.unh.edu/consortiums/iscsi /

[4] Prasenjit Sarkar and Kaladhar Voruganti: “IP Storage: The Challenge Ahead”, Proceedings of Tenth NASA Goddard Conference on Mass Storage Systems and Technologies (2002).

[5] Peter Radkov, Li Yin, Pawan Goyal, Prasenjit Sarkar and Prashant Shenoy: “A Performance Comparison of NFS and iSCSI for IP-Networked Storage”, Proceedings of FAST 2004 pp. 101-114 (2004).

[6] 藤田智成, 小河原成哲: “iSCSI ターゲットソフトウェアの解析”, 先進的計算基盤システムシンポジウム SACSIS (2004).

山口 実靖 Saneyasu YAMAGUCHI

東京大学生産技術研究所 産学官連携研究員．2002 年 東京大学大学院工学系研究科電子情報工学専攻博士課程修了，博士(工学)．iSCSI を用いたネットワークストレージシステムの性能向上の研究に従事．日本データベース学会，情報処理学会正会員．

小口 正人 Masato OGUCHI

お茶の水女子大学理学部情報科学科助教授．1995 年 東京大学大学院工学系研究科博士課程修了，工学博士．ネットワークコンピューティング・ミドルウェアに関する研究に従事．IEEE, ACM, 電子情報通信学会，情報処理学会，日本データベース学会各会員．

喜連川 優 Masaru KITSUREGAWA

1978 年東京大学工学部電子工学科卒．1983 年同大大学院工学系研究科情報工学博士課程修了，工学博士．同年同大生産技術研究所講師．現在，同教授．平成 15 年 4 月より，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.本会理事，情報処理学会フェロー，SNIA-Japan 顧問, ACM SIGMOD Japan Chapter Chair(H11-H14),電子情報通信学会データ工学研究専門委員会委員長(H9,10).VLDB Trustee, IEEE TKDE Assoc. Editor, IEEE ICDE, PAKDD, WAIM Steering Comm.Member.