様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　元 年　 ６ 月 １６ 日現在

機関番号：１５１０１
研究種目：基盤研究(C)（一般）
研究期間：2016～2018
課題番号：１６Ｋ００４７７

研究課題名（和文）P2PモデルとCSモデルを融合したe-Learningシステムの開発に関する研究

研究課題名（英文）Research on development of e-Learning system constructed by integrating P2P model and CS model

研究代表者
川村　尚生（KAWAMURA, Takao）

鳥取大学・工学研究科・教授

研究者番号：１０２６３４８５

交付決定額（研究期間全体）：（直接経費）　　2,600,000円

研究成果の概要（和文）：我々は，高価なサーバ機や大容量ネットワーク設備への投資を行うことなく，多数の学生が同時に使用できる分散型e-Learningシステムの開発を目指している．本研究の主要な成果は，分散型e-Learningシステムのフレームワークを設計・開発したことである．すなわち複数のe-Learningサーバ群に問題データや機能を分散配置し，サーバが動的に参加・離脱できるシステムを，分散ハッシュテーブルを基にして設計した．また，同一コンテンツを多数の学生が同時利用する場合の応答性能を向上させるために，コンテンツを複製してアクセスを分散させる．この複製に関する基礎的な知見が得られたことも本研究の成果である．

研究成果の学術的意義や社会的意義
教育機関におけるe-Learningシステムにおいて，高拡張性を安価に確保するための知見を得た．e-Learningシステムは，学生が自習等に利用するためには高性能は不要だが，動画を用いた研修等を大勢が同時受講する場合には極めて高い性能が必要である．本研究では，安価なパーソナルコンピュータを動的に追加することで，利用者数等の増大に適応できるe-Learningシステムの構成手法を開発した．

研究成果の概要（英文）：We have developed a distributed e-Learning system that can be used simultaneously by many students without investing in expensive server machines and large-capacity network equipment. The main outcome of this study is the design and development of a distributed e-Learning system framework. That is, we distributed the data and functions to multiple e-Learning server groups, and designed the system that the server can join and leave dynamically based on the distributed hash table. Moreover, in order to improve the response performance when the same content is simultaneously used by many students, the content is replicated and access is distributed. It is also the result of this study that basic knowledge about this replication was obtained.

研究分野：社会情報システム

キーワード：e-Learning　分散システム　P2P　モバイルエージェント　分散ハッシュテーブル

１．研究開始当初の背景

現在一般に利用されている e-Learning システムは，どれも学習者が Web ブラウザから単一の Web サーバにアクセスするものであり，図 1-(a)に示すような典型的なクライアントサーバ（以後，CS と略す）モデルに基づいている．しかし，一般に CS モデルでは，サーバに要求が集中するため，以下のような問題が生じる．

(1) サーバの性能が固定されているため，クライアントが多数になるとサーバの負荷増大に伴って学習者の要求への応答時間が長くなる．

(2) サーバが，故障やメンテナンスなど，何らかの原因で停止した場合，システム全体がまったく利用できなくなる．

e-Learning システムに限らず，一般に CS モデル型のシステムでは，高機能かつ高価なサーバ機を導入せずにこれらの問題に対処するために，図 1-(b)のように，複数のサーバを用意してプロキシを導入する方法が利用される．プロキシはクライアントから要求を受けると，適当なサーバを選んで以後の処理を任せる．このとき，負荷が均等になるようにサーバを選択することで，サーバの CPU に対する負荷集中には対処できる．しかしサーバ群の属するネットワークへの負荷集中は軽減されない．また，プロキシという集中的要素がシステムに導入されているため，(2)の問題は残されている．すべてのサーバが同時に停止しない限りサービスは提供できるので，サーバの停止に対する耐性は有しているが，今度はプロキシが停止した場合にシステム全体が利用できなくなってしまう．

研究代表者らは，これまでに，集中的要素を持たず，上記（1），(2）の問題を両方解決する分散型 e-Learning システムについての研究を行ってきた．このシステムでは，学習者が使用するコンピュータは，ブラウザ上に問題を表示して学習者に解答させるだけではなく，システム内に存在する問題のうち一定量を預かり，他の学習者からの要求に応じて送り出す責任を負う．そして，学習者が自習を終えたとき，預かっていた問題を他の学習者のコンピュータに渡す．すなわち，システムを利用中の学習者のコンピュータはすべて，クライアントであると同時にサーバでもある．このシステムは，利用者の増減に応じてサーバ台数が増減するので応答時間が増大しないことが期待でき，一部のサーバが故障しても，残ったサーバによってサービスが継続提供されるので信頼性が高い．すなわち，上記（1），(2）の問題を両方解決したシステムと言える．

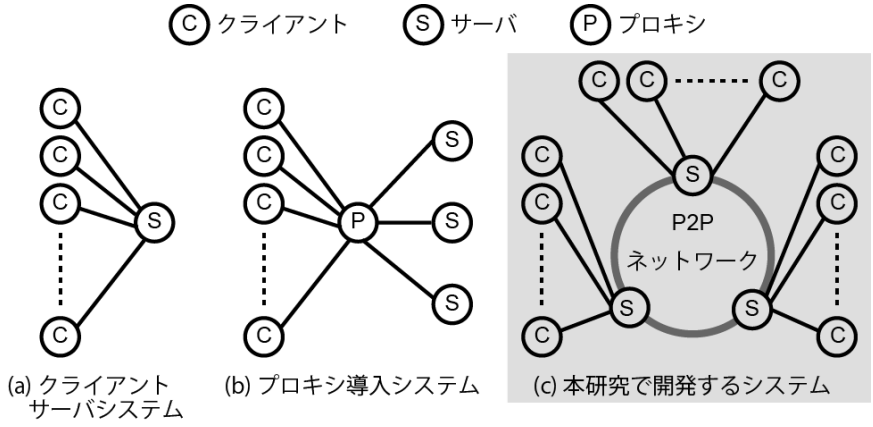

図 1　3 種類のシステム構成

２．研究の目的

本研究以前に，学習者のコンピュータを動的に結合する分散型 e-Learning システムは開発済みで，研究代表者の属する大学の計算機実習室において，情報処理技術者試験の過去問題を，最高 80 名程度の学生に学習させる実証実験を行い，良好な結果を得ていた．

分散型 e-Learning システムは，大学の計算機実習室のような環境での使用には問題ないが，学習者が自宅での自習に使用する場合，特に学習塾や予備校が本システムを導入する場合においては，以下のような問題があった．

(1) 学習者のコンピュータ，ネットワーク環境が不安定で，ネットワーク接続が不意に断たれたり，コンピュータが不用意にシャットダウンされたりすることが懸念される．

(2) 著作権で保護された問題データや，他の学習者の学習記録などの個人情報が任意の学習者のコンピュータ上に一時的にせよ存在することが許容できない．

そこで，本課題においては，分散型 e-Learning システムの技術を基礎として，図 1-(c)に示すような，新たな e-Learning システムの開発に関する研究を行うことを目的とした．このシステムでは，学習者のコンピュータにはサーバの役割を分担させず，学習塾や予備校など e-Learning システムを提供する側のコンピュータ群のみにサーバ機能を提供させることで上記の問題点を解決する．サーバ機能を提供するコンピュータの台数は必要に応じて動的に増減できるようにする．また，サーバ機能を分担するコンピュータとしては，高価な専用サーバ機のみではなく，教員や事務員が使っている PC のような，サービスを提供する組織にもともと存在しているものも利用できるようにする．すなわち，本システムは，ピア・ツー・ピア（以後、P2P と略す）モ

デルと CS モデルを融合し，両者の長所を併せ持つものである．

一般的な e-Learning システムでは，1 台のサーバ機のみがすべての問題データの提供と採点を行うので，同時に学習する学生が増えるほどサーバ機の負荷が増大し，応答速度が低下してしまう．その点，提案する e-Learning システムでは，サーバ機能を分担するコンピュータを動的に必要なだけ増減できる点が優れている．一般に，サーバ機の性能はピーク時に必要とされる能力を上回るように設定する必要があり，多くの時間帯においては過分な性能を持て余すことになる．すなわち，サーバ機に対する投資効率はあまり高くないと言える．その点，本研究で開発する e-Learning システムは，たとえば教員や事務員用の有休 PC に一時的にサーバ機能を分担させることができ，常に必要にして十分な性能を発揮するようにシステムを運用できる．

本研究で利用する P2P モデルや CS モデルなどのアイデアは既存のものだが，それらを組み合わせ，e-Learning システムを分散型システムとして実現したものは実用化されていない．基礎技術を実用化するには様々な現実的問題を解決する必要があり，本研究の意義はその点に関する知見を得ることにある．

３．研究の方法

(1) 分散型 e-Learning システムの基本的な仕組み

e-Learning システムにおいて，複数のサーバが問題データを分担して管理し，しかもサーバを動的に増減するためには，現在どのサーバがシステムに参加しており，どのサーバがどの問題データを保有しているかに関する情報を管理しなければならない．これらの情報を単一の集中的なデータベース管理システムに格納する一般的な手法を採用すれば，そこが単一障害点となり，分散型システムを構築する意義が失われてしまう．したがって，このデータベース自体を分散型にする必要がある．これには，従来から研究開発が行われている分散ハッシュテーブルを用いる．

分散ハッシュテーブルは，アドホック性とスケーラビリティの両立を目指す探索手法で，アドレスとコンテンツのハッシュ値を空間に写像し，その空間を複数のノードで分割管理することで，特定ノードに負荷が集中することなく大規模なコンテンツ探索を実現する．具体的には，key と value の組を複数個格納し，key に対応する value を参照するためのデータ構造であるハッシュテーブルを，単一のノードではなく，複数のノードで分散管理する．ここで，key とは，特定の情報を検索する上で必要となる情報を表し，value とは，key と関連付けられた情報を表す．分散ハッシュテーブルの実装アルゴリズムは複数存在するが，そのすべてに共通する API として，put（ハッシュテーブルに指定した key を主キーとして value を挿入・更新する），get（ハッシュテーブルから指定した key を検索し value を取得する），join（ノードネットワークにノードを参加させる），leave（ノードネットワークからノードを離脱させる）がある．この join と leave を利用することによって，サーバを動的に追加・離脱させることができる．

分散型 e-Learning システムに参加しているサーバに関するデータは，(1)サーバ ID (MAC アドレス)，(2)サーバの IP アドレス，(3)サーバのディスク残量の 3 つ組からなり，これらを分散ハッシュテーブルによって管理する．ディスク残量は，新しい問題データを配置する際に参考情報として用いる．

問題データについては，当初は分散ハッシュテーブルにそのまま格納することを計画していたが，動画などの巨大な問題データをそのままデータベースに格納するのは容量制限から難しいことと，サーバの離脱時に巨大なデータを他のサーバに移動させるのに要する時間が無視できないことがわかった．また，教育組織に既存の遊休コンピュータやファイルサーバを活用するなど，ストレージに関しては柔軟な提供体制を構築した方がよいと考えた．そこで，分散データベースには問題データへの参照だけを格納し，実際の問題データは別途用意するストレージシステムに格納することとした．ここで，問題データの参照とは，(1)問題データ ID，(2)問題データを実際に格納しているコンピュータの ID アドレス，(3)当該コンピュータ上のパスの 3 つ組からなり，これらを分散ハッシュテーブルによって管理する．

設計したシステムが問題なく動作することを確認するために，オープンソースの分散データベース管理システムである Apache Cassandra を，我々が開発したモバイルエージェントシステムプラットフォーム Maglog にモジュールとして組み込み，プロトタイプシステムの実装を行って動作確認を行った．Apache Cassandra は単一障害点を持たず，動的にノードの追加・削除が行えるため，本研究の分散型 e-Learning システムの基盤要件に適合している．なお，Maglog は Java 上で動作するプラットフォームで，Java 言語や Prolog 言語を用いて，エージェントベースのアプリケーションを記述できる．

(2) ストレージシステム

前節で述べたように，分散型 e-Learning の問題データを格納するストレージシステムを実現することが必要である．分散システムなので，やはりストレージシステムも分散型としたい．そこで，既存の分散ストレージシステムの一種である，オブジェクトストレージシステムを使用することとした．オブジェクトは保存場所を示す固有の識別子と，様々な情報を記録可能なメタデータが付与された上で，階層構造ではない平坦な空間に格納される．固有の識別子が実際に保存されるディスク上の場所に依存しないことで，データの配置に制約が少ない．このため，データの移動や分散配置が容易である．大学には，機器の更新などにより用途がなくなり放置や廃棄されてしまう，本来であれば継続して利用可能な使用済みのコンピュータが多数存在するので，そ

れらを有効活用することを考えた．使用済みコンピュータを分散ストレージとして再利用する上で必要な作業を全て手動で行うと，多大な労力を要してしまう．このため，使用済みコンピュータに分散ストレージとして利用するための環境を自動構築するシステムを，既存のソフトウェアを組み合わせて，以下の通り実装した．

① 【OS の自動インストール】　クライアントである使用済みコンピュータから PXE ブートの要求が行われると，DHCP サーバが前述のスクリプトにより固定 IP アドレスを払い出す．その後，TFTP サーバからブートイメージが，HTTP サーバから OS のイメージや Kickstart 関連のファイルが読み込まれることで OS の自動インストールが実行される．これにより，使用済みコンピュータをネットワークに接続して PXE ブートさせる作業のみで，OS のインストールが自動的に完了する．

② 【ストレージソフトウェアの自動インストール】　オブジェクトストレージを構築するためのソフトウェアとして，OpenStack Swift を利用した．オブジェクトはコンテナで，コンテナはアカウントで一覧を管理される．アカウントを担う Account Server，コンテナを担う Container Server，オブジェクトを格納する Object Server はまとめてストレージノードと呼ばれ，ゾーンという単位でグループ化される．ゾーンは複数作成することができ，それぞれに同じアカウント，コンテナ，オブジェクトが格納されるため，冗長化や耐障害性の向上を実現できる．また，ストレージノードへの通信はプロキシノードと呼ばれる Proxy Server が中継する．Swift の自動インストールには Ansible を利用した．Swift のインストールに必要な設定を記述した Playbook などを作成や配置することで，使用済みコンピュータにオブジェクトストレージとして利用するための環境が自動的に構築される．

この自動構築システムにより構築処理を行った使用済みコンピュータ（CPU：Core 2 Duo E7400 2.8GHz，メモリ：2GB，NIC：1Gbps，HDD：160GB（1 ディスク））15 台を合計容量 72TB のオブジェクトストレージシステムとして構築し，本研究における実験に利用した．今回は，使用済みコンピュータをストレージシステム専用機として利用する方法を開発したが，維持には電気代がかなりかかることがわかった．今後，使用済みではなく，業務に利用中のコンピュータの余剰ストレージを活用することを検討したい．

(3) 本システムを用いた学習

図２に，本研究で開発した e-Learning システムの構成要素と，利用者が本システムを用いて学習するときの処理の流れを示す．

サーバ群は，保守時以外は常に存在するコアサーバ群と，利用者の増減に応じて動的に参加・離脱する遊休サーバ群に大別される．学習コンテンツはすべてのサーバ群に分散配置される．遊休サーバ群が参加・離脱する際には，必要に応じて，コンテンツの複製が配置されたり，削除されたりする．利用者はコンピュータ，タブレットのウェブブラウザから，本システムにアクセスする．

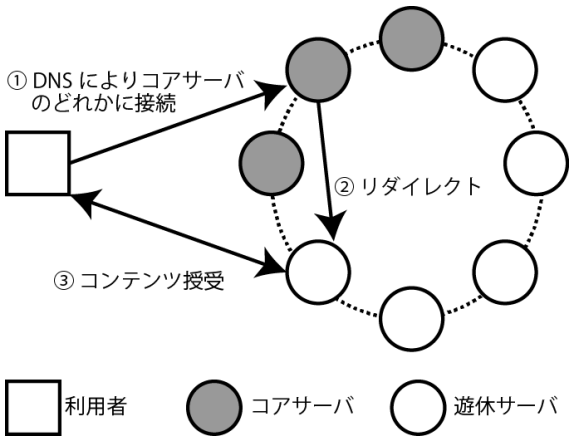

図 2　システムの構成要素と処理の流れ

利用者が本システムを用いて学習するときの処理の流れは以下の通りである．なお，下記箇条書きの番号は図２の矢印付近に書かれた説明の番号と対応している．

① 利用者が使用するウェブブラウザが，e-Learning システムを構成するコアサーバのいずれかにアクセスする．ここでは，特定のサーバではなく，コアサーバのいずれかとする．コアサーバのうちの特定のひとつではなく，コアサーバのいずれかにアクセスすることの実現には DNS を利用する．DNS は www.tottori-u.ac.jp 等の名前を 160.15.14.78 等の IP アドレスに変換する仕組みだが，DNS には 1 つの名前に複数の IP アドレスを登録しておき，問い合わせごとにその中から 1 つの IP アドレスを選んで返す機能がある．コアサーバ群をすべて登録しておくことで，負荷分散が図られると同時に，保守等で特定のサーバが停止している場合でもシステム全体のサービスは継続することになる．

② コアサーバのいずれかが，すべてのサーバの中から適切な対応サーバを選んで，処理をリダイレクトする．図２では遊休サーバの 1 つが対応サーバになっているが，コアサーバが対応サーバになることもある．

③ 以降は，利用者のウェブブラウザと対応サーバが直接コンテンツをやり取りする．DNS が指定したコアサーバ自身ではなく，処理をリダイレクトされた対応サーバがコンテンツを提供することで，さらなる負荷分散を図る．ただし，対応サーバが利用者の求めるコンテンツを保持しているとは限らない．たまたま所持している場合は直接送り出し，保持していない場合は，そのコンテンツを保持しているサーバを探してコンテンツを受け取り，利用者に送り出す．

上記のように，学習にはウェブブラウザを利用するが，特にスマートフォンやタブレットにおいては，ブラウザのエンジンをラッピングしたアプリケーションを用意した方が便利かもしれない．この検討は今後の課題として残されている．

(4) 問題データの複製

問題データは複数の複製が必要である．第一の理由は可用性を高めることにある．ストレージシステムに問題データがただ一つしかない場合，それを格納しているストレージノードが保守やトラブルのため停止中の場合，その問題データを用いた学習ができなくなる．バックアップはストレージシステム外に存在するようにすべきではあるが，その場合でも，システムに参加しているサーバに問題データが保有されていない限り学習ができない点には変わりがない．また，性能を高めるためにも複製は必要となる．同じ問題データを複数の学習者が同時に学習する場合，アクセスを分散させた方が応答速度は上がると考えられる．

可用性と応答速度を高めることだけを考えれば，複製は多ければ多いほどよく，すべてのストレージノードがすべての問題データを格納することが最良の状態となる．しかし，問題データの複製が多ければ，各ストレージノードが持つ問題データの総和も大きくなり，ストレージノードの減少時にそれらの問題データを移動させなければならないので，結果として応答速度が下がることになる．また，各ストレージノードのストレージ容量は有限であることも注意しなければならない．

そこで，複製を 2 種類設けることにした．一方は，ストレージノードの減少時に移動すべき複製（強い複製と呼ぶ）であり，もう一方は，キャッシュのように，ストレージノードの減少時には単に破棄し，増加時には無視される複製（弱い複製と呼ぶ）である．強い複製は，システムで予め決めておいた数だけ，自動的に生成する．実装に用いた Apache Cassandra ではレプリケーション係数を指定することで強い複製を生成できる．弱い複製は，問題データの利用パターンに基づき，必要に応じて生成され，適切なストレージノードに配置される．どのような場合に弱い複製を生成するかは分析の余地があるが，今回の実験では，ノードごとに問題データがリクエストされた数を記録しておき，それがしきい値を超えた場合に生成することにした．ただし，一定時間ごとにリクエスト数はクリアする．すなわち，LRU を近似したアルゴリズムとなっており，最近よく使用される問題データの複製が作られることになる．弱い複製を作成するノードは残容量の大きいノードとしたが，弱い複製はキャッシュのようなものなので，ネットワーク的に近傍のものを利用すると応答速度が上がると考えられる．各ノードがどの程度の頻度で問題データがアクセスされたときに複製を生成し，どこに配置すれば応答速度が最もよくなるかといった分析については本研究では十分調べられなかったので，今後の課題としたい．

４．研究成果

我々は，高価なサーバ機や大容量ネットワーク設備への投資を行うことなく，多数の学生が同時に使用できる分散型 e-Learning システムの開発を目指している．この e-Learning システムは，学習者が多数になっても応答速度が低下しないよう，動的に必要なだけの性能を確保すること，サーバの一部が故障したり，サーバの一部をメンテナンスのために停止させたりしても，システム全体としてはサービスを途切れなく提供し続けられることを目指し，複数台のコンピュータに問題データや機能が分担された分散システムとして設計している．

本研究の主要な成果は，分散型 e-Learning システムのフレームワークを設計・開発したことである．すなわち複数の e-Learning サーバ群に問題データや機能を分散配置し，サーバが動的に参加・離脱できるシステムを，分散ハッシュテーブルを基にして設計した．設計したシステムが問題なく動作することを確認するために，オープンソースの分散データベース管理システムである Apache Cassandra を，我々が開発したモバイルエージェントシステムプラットフォーム Maglog にモジュールとして組み込み，プロトタイプシステムの実装を行って良好な実験結果を得た．Apache Cassandra は単一障害点を持たず，動的にノードの追加・削除が行えるため，本研究の分散型 e-Learning システムの基盤要件に適合している．なお，Maglog は Java 上で動作するプラットフォームで，Java 言語や Prolog 言語を用いて，エージェントベースのアプリケーションを記述できる．この分散ハッシュテーブルには，サーバやユーザの情報および問題データへの参照が格納される．問題データの実体は分散ストレージの一種であるオブジェクトストレージを用いる．プロトタイプシステムではオブジェクトストレージ基盤として OpenStack Swift を利用している．

学生が使用する，パーソナルコンピュータ，タブレット，スマートフォンなどのクライアント端末から，e-Learning システムに接続して学習する枠組みも検討した．負荷状況を考慮したラウンドロビンアルゴリズムにより，特定のサーバにクライアントからの接続要求が集中しないように設計を行った．問題データ，サーバ・クライアント間を HTML でやりとりされ，学習者はウェブブラウザを通じて学習する．

学生が増加した場合に，十分な応答性能を確保するためには，まずサーバ数を増やすことが必要で，それが自動的に行われることが本研究の第一の成果だが，それだけでは，同一問題データを多数の学生が同時利用する場合の応答性能は得られない．この場合は問題データを複製してシステム内に複数存在させ，アクセスを分散させる必要がある．この複製に関する知見が得られたことも本研究の成果である．一般に，分散システムでは，データの複製はシステムによって管理されており，システム全体に分散して存在する．複製の存在により，同時利用時の応答性能が確保できるだけでなく，個々のサーバがクラッシュした場合でも，システム全体としては途切れなく問題データを提供できることになり，可用性も高まる．複製は多ければ多いほど応答性能や可用性が高くなるが，本研究の分散型 e-Learning システムにおいては，サーバが離脱する際に

は保持している問題データの移動が必要となるので，複製があまりに多く存在すると，それに要する時間が多大なものとなってしまう．通常の複製（強い複製と呼ぶ）とは別に，サーバが離脱する際に移動させず単に破棄するキャッシュのような複製(弱い複製と呼ぶ)を導入することで，システム全体の性能があげられることがわかった．強い複製，弱い複製をどの程度の割合で，どのサーバに配置すると最も性能が上がるかといった分析は今後の課題として残されている．

５．主な発表論文等

〔雑誌論文〕（計　2 件）

(1) Toshiya Kawato, Shin-ichi Motomura, Masayuki Higashino, Takao Kawamura, Auto-Construction for Distributed Storage System reusing Used Personal Computers, Journal of Computers, Vol. 13, No. 10, pp. 1156-1163 (2018), doi: 10.17706/jcp.13.10.1156-1163, 査読あり

(2) Toshiya Kawato, Shin-ichi Motomura, Masayuki Higashino, Takao Kawamura, Attempt to Reuse Used-PCs as Distributed Storage, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 12, No. 2, pp. 93-96 (2018), https://waset.org/publications/10008791/attempt-to-reuse-used-pcs-as-distributed-storage, 査読あり

〔学会発表〕（計　5 件）

(1) Toshiya Kawato, Masayuki Higashino, Kenichi Takahashi, Takao Kawamura, Proposal of e-Learning System integrated P2P Model with Client-Server Model, The 18th International Conference on Electronics, Information, and Communication, pp. 328-333 (2019).

(2) Toshiya Kawato, Shin-ichi Motomura, Masayuki Higashino, Takao Kawamura, Auto-Construction for Distributed Storage System reusing Used Personal Computers, Proceedings of the 2018 6th International Conference on Network and Computing Technologies, pp. 1-8 (2018).

(3) Toshiya Kawato, Shin-ichi Motomura, Masayuki Higashino, Takao Kawamura, Attempt to Reuse Used-PCs as Distributed Storage, Proceedings of the ICCNC 2018: 20th International Conference on Computer Networks and Communications, pp. 426-429 (2018).

(4) 川戸 聡也, 本村 真一, 東野 正幸, 川村 尚生,使用済みパソコンを再利用した分散ストレージシステム, 情報処理学会第 80 回全国大会講演論文集, Vol.1, pp. 23-24 (2018).

(5) 東野 正幸, 川村 尚生,論理型モバイルエージェントフレームワークの開発とマイクロサービスアーキテクチャとの調和, 合同エージェントワークショップ＆シンポジウム 2016 予稿集（JAWS 2016）, pp. 235-236 (2016).

６．研究組織

(1) 研究分担者

研究分担者氏名：菅原　一孔

ローマ字氏名：SUGAHARA kazunori

所属研究機関名：鳥取大学

部局名：工学研究科

職名：教授

研究者番号（8 桁）：90149948

研究分担者氏名：高橋　健一

ローマ字氏名：TAKAHASHI kenichi

所属研究機関名：鳥取大学

部局名：工学研究科

職名：准教授

研究者番号（8 桁）：30399670