

PrinterSurf:クラウド型プリンタ共有システムの開発

吉田忍^{†1} 西岡大^{†1} 村山優子^{†1}

近年、様々なネット印刷サービスやプリンタ共有システムが登場し、手元にプリンタが無い状態での印刷が可能となりつつある。しかし、既存のサービスやシステムでは、プリンタの利用、提供が容易に行われているとは言い難い。本研究では、誰でも容易にプリンタを公開、利用可能なシステム PrinterSurf を開発した。本稿では、大規模な利用に対応したクラウド型 PrinterSurf の開発を報告する。

PrinterSurf: An implementation of a Printer Sharing System based on Cloud

SHINOBU YOSHIDA^{†1} DAI NISHIOKA^{†1} YUKO MURAYAMA^{†1}

In recent years, we have various net printing services as well as the PrinterShare system. With these technologies, we can print documents even if no printer is locally available. However, existing systems and services are often neither easy to use nor to provide. In this research, we developed a printer sharing system PrinterSurf. With this system, anyone can use and provide printer easily. This paper reports on our implementation of PrinterSurf based on Cloud for a large-scale use.

1. はじめに

近年、情報機器の小型化や、無線アクセスポイントの増加等により、資料の作成や編集、情報収集等を場所を選ばずに行うことが可能となった。また、モバイル環境下では利用の難しかった印刷デバイスに関しても、様々なネットワーク印刷サービスやプリンタ共有システムが登場し、徐々に利用可能となりつつある。しかし、既存のサービスやシステムでは、利用、公開双方において容易に行えるものは少なく、専門的な知識が必要になる場合も多い。

我々はこれまで、利用、公開双方をユーザによって容易に実現できるユビキタス印刷サービス PrinterSurf の実装を行ってきた。PrinterSurf システムはプリンタ提供者とプリンタ利用者によって成り立っている。プリンタ提供者がデバイスや消耗品の提供とドライバのインストールや設定を肩代わりする事により、利用者はネットワークに接続するだけで印刷サービスが利用可能となる。先行研究[1]では、P2P 型のネットワークで PrinterSurf を実装し、運用と評価を行った。運用評価のユーザアンケートにおいて、「システムが便利である」、「今後も使ってみよう」という意見が過半数であったことから、モバイル環境下での印刷サービスにはニーズが存在し、はその解決手段として有効であるといえる。

しかし、P2P を使用することに対して情報漏洩の不安やデータの送信経路の不透明性等から使用をためらうケース

も存在した。また、システムを JAVA で実装したため、を利用するには、PC に JAVA を使用するための VM をインストールしなければならないという問題が存在した。

この問題を解決するため、我々はシステムをクライアントサーバ型で再構築し、プリンタ利用を Web ブラウザから行うことで、JAVA のインストールを不要とした[2]。しかし、クライアントサーバ型では、サーバへ負荷が集中してしまう。そのため、アクセスの増加に伴い、サーバの処理速度や帯域がネックになることが予想される。本研究では、サーバの負荷を軽減するため、クラウドサーバを構築し、その上でサーバの機能を分散させ、システムの耐負荷性の向上を図る。

2. 関連研究

本節では、の関連研究として、類似したネットワークプリントサービスを挙げ、との比較を行う。

2.1 ネットワークプリントサービス

本節では、と同様に、ネットワーク上のプリンタを利用するサービスの例を挙げる。

2.1.1 IPP(Internet Printing Protocol)

IPP[3][4] は IETF によって標準化されている印刷プロトコルある。IPP 対応プリンタであればネットワークを介してプリンタの共有が可能である。しかし、IPP 対応プリンタでなければ使用できず、利用者自身がドライバをインストールする必要がある。そのため、プリンタの利用、公開の双方に専門的な知識が必要である。

2.1.2 ネットプリント(net print)

ネットプリント[5] は富士ゼロックスによって提供されている印刷サービスである。利用者はパソコンやスマート

^{†1} 岩手県立大学大学院ソフトウェア情報学研究科

フォンなどから印刷物を登録し、予約番号を取得、コンビニエンスストアに設置されたマルチコピー機に番号を入力することで印刷が行われる。インストール作業は不要であり、簡単に利用可能であるが、利用可能なプリンタはコンビニエンスストアに設置してあるものに限られるため、個人がプリンタを提供するのは難しい。

2.1.3 UPnP(Universal Plug and Play)

PnP はコンピュータに周辺機器を接続した際に、OS が自動的に機器を検出して組み込みと設定を行うシステムである。UPnP[6] は PnP の概念をネットワークまで拡大したもので、パソコンや周辺機器をネットワークを通じて接続し、機能を利用するものである。これによりネットワーク上のプリンタを接続するだけで共有可能となるが、広域ネットワークでの利用は想定されておらず、小型オフィスや家庭などの小規模ネットワーク向けのシステムである。

2.1.4 ThinPrint

ThinPrint[7] はコルダート社(Cortado AG) が提供するシンクライアント環境向けの印刷ソリューションである。利用者はネットワークに接続されている端末から自由に印刷可能であるが、ThinPrint を利用するには専用のサーバを設置せねばならず、容易に公開が出来るとは言いがたい。また、シンクライアント向けのシステムであるため、個人が利用するのは難しい。

2.1.5 PrinterOn

PrinterOn[8] は PrinterOn Corporation が提供するモバイル印刷サービスである。登録されたプリンタへ対応する E メールアドレスを割り当て、そのアドレスに対してドキュメントを送信、印刷を行う。スマートフォンやタブレット、PC など幅広い機器からの利用が可能であり、様々な利用者向けに多様なサービスを提供している。しかし、個人でプリンタを公開する場合、公開できる相手の人数制限が存在する。利用者が指定できるプリンタは殆ど図書館や教育機関など公的なものが多い。

2.2 印刷サービスの比較

ネットワークを利用した印刷サービスについて、プリンタ利用の容易さ、公開の容易さ、対象機器、ネットワークの範囲についての比較を行う。比較の結果を表 1 に示す。

他サービスと比較して、PrinterSurf は Web ブラウザから利用でき、公開する際にも煩雑な作業がいらぬため、利用、公開の 2 点両方が優れているといえる。プリンタ提供者がボランティアでプリンタを公開し、ネットワーク上で広くプリンタデバイスを共有するシステムである。そのため、UPnP や ThinPrint, PrinterOn とは利用モデルが異なる。

表 1 : サービスの比較

	プリンタ利用の 容易さ	プリンタ提供 の容易さ	対象機種	ネットワーク
	○	○	PC	WAN
IPP	×	×	PC	WAN
ネットプリント	○	×	PC, スマートフォン	WAN
UPnP	○	○	PC	ホーム ネットワーク
ThinPrint	○	×	シン クライアント	WAN
PrinterOn	○	△	PC, タブレット, スマートフォン	WAN

3. PrinterSurf

本節では、PrinterSurf の概念と先行研究、先行研究の問題点について述べる。PrinterSurf では、プリンタ提供者が、システムへプリンタを公開する。プリンタ利用者は システムを介し、任意のプリンタへ印刷を行う。システムモデルを図 1 に示す。

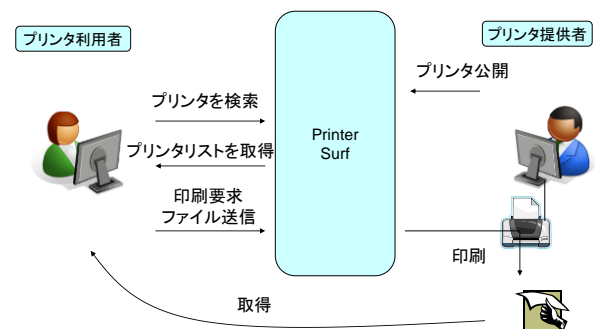


図 1 : システムモデル図

利用に際してドライバのインストールや設定は不要であり、登録されているプリンタならばどれでもそのまま利用できる。プリンタ提供者は自分が利用可能なプリンタであれば公開可能であるため、新規にプリンタを用意する必要はない。これらの事から容易に公開、利用が可能であるといえる。

3.1 マルチキャスト型 PrinterSurf

高橋ら[9]は、実社会における紙媒体の有用性に着目し、通信相手のプリンタに直接印刷することを目的とした、PrinterSurf システムの提案を行った。実装されたシステムでは、ネットワーク上のプリンタを発見するためにマルチキャストを使用している。そのため、プリンタの利用者と利用対象となるプリンタは同一ネットワーク内である必要があった。

3.2 ハイブリッド P2P 型 PrinterSurf

中上ら[10]は、プリンタの効率的な利用、発見を行うために、ハイブリッド P2P 型でシステムの実装を行った。このシステムでは、インデックスサーバに保管された情報から対象のピアを発見、その情報を元にピア間でファイルの送受信を行い、印刷される。しかし、高橋らと同様に同一ネットワーク内の利用に限定されている。

3.3 ピュア P2P 型 PrinterSurf

齊藤ら[1]は、インデックスサーバを廃し、各クライアントが同一の機能をもつピュア P2P 型としてシステムを実装した。このモデルでは、ハイブリッド P2P 型でインデックスサーバが持っていた情報を各ピアが所持している。また、上位のネットワークに存在するピアを通じてオーバーレイネットワークを構築することにより、同一ネットワーク外にあるプリンタの利用を可能とした。しかし、同時期に P2P ファイル共有ソフトが問題になったことによる P2P 技術に対する不信感や、システムを JAVA で実装したため、新規に JAVA のインストールが必要になった場合、企業の PC 等ではセキュリティ上の規約により、新規に JAVA のインストールができない場合があるといった問題がある。

3.4 クライアントサーバ型 PrinterSurf

我々は、ピュア P2P 型での問題点を解決するため、クライアントサーバ型で PrinterSurf システムの実装を行った[2]。ネットワークアーキテクチャの変更だけではなく、システムを Web ブラウザから利用する形態とし、JAVA のインストールも不要となった。これによって、先行研究の問題点は解決できたとと言える。しかし、各クライアント間の処理をサーバが行うため、サーバに負荷が集中するという新たな問題が発生した。

3.5 先行研究のシステム概要

本節では、先行研究のクライアントサーバ型 PrinterSurf の概要について述べる。PrinterSurf システムは PrinterSurf サーバ、プリンタ提供者クライアント、プリンタ利用者クライアントの3つで構成されている。クライアントサーバ型のシステム構成を図2に示す。PrinterSurf サーバ、プリンタ提供者クライアントに JAVA を用いて、プリンタ利用者クライアントに JAVAServlet を用いて開発を行った。Web サーバには Apache Tomcat を使用した。

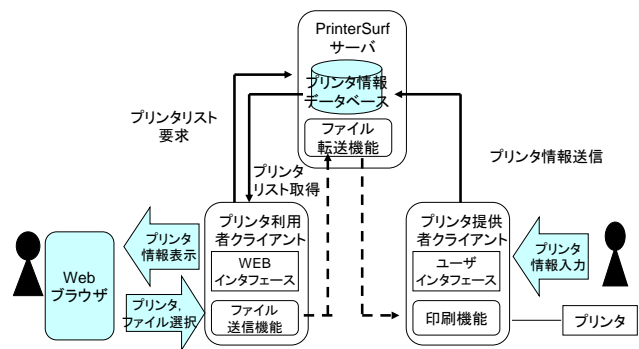


図2：システム構成

3.5.1 PrinterSurf サーバ

PrinterSurf サーバは、プリンタ提供者クライアントから送られてきたプリンタ情報の管理と、プリンタ利用者クライアントとプリンタ提供者クライアント間でのデータ転送を行う。プリンタ情報にはプリンタの名前、プリンタの所有者、プリンタの場所等が記載される。各情報はプリンタ提供者がプリンタ提供者クライアントから自由に書き込むことが出来る。プリンタ情報を受信すると、PrinterSurf サーバは、受信した情報を元にプリンタ情報データベースの構築を行う。データベースには SQLite を用いた。

3.5.2 プリンタ提供者クライアント

プリンタ提供者クライアントは PrinterSurf サーバへのプリンタ情報の登録、PrinterSurf サーバから送信されたファイルの受信と印刷を行う。印刷に対応しているフォーマットは PDF, Microsoft office Word, PowerPoint である。プリンタ提供者クライアントは PC にインストールされているプリンタドライバを取得し、プリンタ提供者はその中から公開するプリンタを選択する。プリンタの所有者や場所などの情報を入力し PrinterSurf サーバに送信、登録が完了するとプリンタ提供クライアントは印刷待機状態となり、ファイルを受信するたびにその印刷を行う。なお、プリンタドライバ情報の取得に Win32DLL, 印刷の実行に WHS を用いている為にプリンタ提供者クライアントのコンピュータは Windows である必要がある。

3.5.3 プリンタ利用者クライアント

プリンタ利用者クライアントは Web クライアントであり、プリンタ利用者は Web ブラウザを用いてシステムを利用する。プリンタ利用者クライアントは、PrinterSurf サーバのデータベースの内容を取得し表示する。プリンタ利用者はその中から使用したいプリンタを選択し、ファイルをアップロードする。アップロードに際し、形式によっては印刷の設定を行う画面を表示する。アップロードが終了すると、PrinterSurf サーバを介してプリンタ提供者の元へ送られる。

3.6 クライアントサーバ型の問題点と解決策

クライアントサーバ型で構築したシステムにおいて、多数のアクセスがサーバに集中することにより、通信速度の上限によってファイル転送に支障がでることと、Webサーバとデータベースに負荷がかかりダウンすることが考えられる。例として、富士ゼロックスのネットプリントでは、全国のコンビニ約 15000 店舗にプリンタが設置されているが、それをすべて 1 台のサーバで管理するのは困難である [5]。解決策として、1 台のサーバの性能と通信速度を高めるか、サーバ機能を分散し複数台で運用する方法が挙げられる。1 台のサーバには性能の限界があるが、複数台で運用する場合はリソースの消費にあわせてサーバ数を増加させることで、より多くのアクセスに対応できると考えられる。しかし、複数の物理サーバを用意するには非常に手間とコストがかかってしまう。そこで、低コストかつ容易に複数の仮想サーバを使用できるクラウドサーバについて調査を行った。

4. クラウド化の検討

本節では、クラウドコンピューティングの概念と特徴を述べ、代表的なクラウドサービスを挙げる。また、本研究で使用するクラウドミドルウェアについて考察する。

4.1 クラウドコンピューティング

NIST(National Institute of Standards and Technology)によるクラウドコンピューティングの定義[11]では、クラウドとは、ユーザがクラウド上のリソースを利用する際に、リソース資源の物理的な場所を意識せず、管理者と直接やり取りすることもなく、システムによって自動的にリソース使用量の最適化が迅速に行われ、かつ様々なプラットフォームから利用できるものとされている。実際のクラウドサーバ上では、1 台の物理サーバ内に複数の仮想サーバを起動し、ユーザはネットワーク越しに各仮想サーバへアクセスする。これによってリソースの有効活用が可能である。

実装モデルとして、プライベートクラウド、コミュニティクラウド、パブリッククラウド、ハイブリッドクラウドの4つが挙げられている。プライベートクラウドは利用する組織内に設置され、その組織のためだけに利用される。コミュニティクラウドでは、インフラはいくつかの、共通の目的をもつ組織によって利用される。パブリッククラウドのインフラは広く公開される。パブリッククラウドの管理はサービスを販売する組織によって行われる。ハイブリッドクラウドは、複数のクラウドの組み合わせで構成される。各クラウドは独立しているが、データやアプリケーションは移動可能である。

4.2 クラウドサーバの特徴

クラウドコンピューティングを用いたクラウドサーバの特徴として、物理サーバにとらわれることなく、仮想サーバで動的にサーバ数の増減をすることが可能であること

があげられる。人的、設備的コストをかけずにサーバの規模の拡大、縮小が行えるため、従来の物理サーバに比べ、アクセスの一時的な増加への対応や、システムの規模拡大が非常に容易である。また、ソフトウェア側から要求に自動的に対応してサーバ数を増減させることができる。これをオートスケーリングと呼ぶ。

4.3 商用クラウドサービス

本節では、代表的な商用クラウドサービスを挙げ、その特徴を述べる。

4.3.1 Amazon EC2/S3[12]

Amazon が提供する IaaS 型のサービス。Linux や CentOS, Windows Server の環境をクラウドサーバ上で利用できる。サーバの環境をそのまま使えるために自由度が高い。サービスの規模に合わせてクラウドサーバの機能を同的に変更することができる。2013 年時点でのシェアは最も高い [13]。

4.3.2 Google Apps Engine[14]

Google が提供する PaaS 型のサービス。Python, JAVA, Go の実行環境と Google 独自のデータベースである BigTable が利用できる。高速かつ大規模な BigTable を利用できる反面、環境にあわせたプログラムの修正が必要であり、実行できるアプリケーションに限られる。

4.3.3 Force.com[15]

Salesforce.com が提供する PaaS 型サービス。Salesforce 側から提供される多数のソリューションが存在するメリットがある。しかし、独自言語を使用せねばならないため、既存のシステムをそのまま公開するのは難しい。

4.3.4 Windows Azure[16]

Microsoft が提供する PaaS 型サービス。他の商用クラウドサービスに比べ、仮想マシンとして Windows Server が最初から用意されており、環境もそれに特化しているという特徴がある。

4.3.5 IBM クラウド[17]

IBM が提供する IaaS 型サービス。独自の機能として、クラウドサーバ上の仮想サーバに加え、物理サーバを取得することができる。

4.4 クラウドミドルウェア

現在、様々な企業や研究機関より、オープンソースのクラウドミドルウェアが公開されている。クラウドサーバの構築を行うにあたり、代表的なクラウドミドルウェアをいくつか抜粋した。以下にその特徴を述べる。

4.4.1 Eucalyptus[15]

Eucalyptus Systems によって管理されている。IaaS 機能を提供する。仮想化は Xen, KVM に対応しており、Linux 上で動作する。商用クラウドである Amazon EC2/S3 と互換性がある。オートスケール機能に対応している。

4.4.2 OpenNebula[16]

マドリード・コンプルテンセ大学で開発され、現在は OpenNebula Project によって管理されている。仮想化は Xen,

KVM の他 VMware にも対応している。Linux だけでなく、MacOS X でも動作する。

4.4.3 OpenStack[17]

Rackspace Cloud Computing と NASA が共同で開発を行っている。仮想化は Xen と KVM。大規模なインスタンス作成に対応している。

4.4.4 ミドルウェアの選定

本研究では、まずプロトタイプシステムの実装を行う環境として、プライベートクラウドの構築を行う。システムの完成後、パブリッククラウドへ移行し、公開を行う予定である。そのため、プライベートクラウドの構築に使用するミドルウェアを選定するにあたり、JAVA が動作すること、商用クラウドと互換性があることが望ましい。そこで、仮想マシン上で Linux が動作し、環境をユーザが構築できるため JAVA やその他のライブラリを使用でき、AmazonEC2/S3 との互換性があり、オートスケールに対応している Eucalyptus を使用することとした。

5. 提案システム

本節では、クラウド化を行う箇所の考察と、システムの概要、各サーバ、クライアントの機能を述べる。

5.1 クラウド化を行う箇所

クラウドサーバは、容易にサーバ数を増減させることが可能であり、システムの規模の変更に容易に対応できる。また、オートスケール機能により、サーバの起動と消去を人の手を介することなくソフトウェアから行うことができる。そのため既存のサーバに比べ、非常に低コストで分散型のシステムを構築し、運用することが可能である。本稿では、先行研究のサーバをプリンタ管理サーバとファイル転送サーバに分割し、クラウドサーバ上に配置する。また、ファイル転送サーバの台数を、負荷によって増減させることで、1 台のサーバあたりの接続数の減少を図る。

5.2 システム概要

本システムは、利用ユーザに Web インタフェースを提供するプリンタ利用者クライアント、プリンタ情報の管理と保管を行う管理サーバ、印刷対象のファイル転送を行う転送サーバ、プリンタ公開を行うプリンタ提供者クライアントの 4 つで構成されている。システム構成を図 3 に示す。本稿では、管理サーバについてはスケールを行わず、転送サーバのみのスケールを行う。

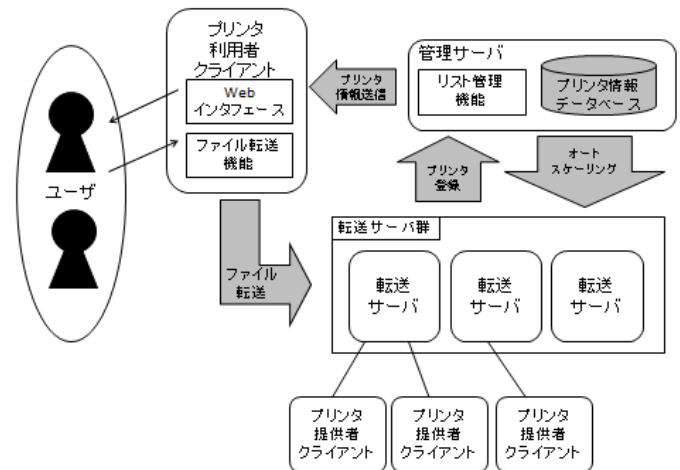


図 3：システム構成

5.2.1 管理サーバ

管理サーバは登録されているプリンタ情報をデータベース化して保管し、要求に応じて Web サーバを通してユーザに提供する。また、転送サーバの状態を監視し、オートスケールを行う。プリンタ情報データベースには、従来のプリンタの名前、プリンタの所有者、プリンタの場所に加えて、プリンタ自身の ID とプリンタ提供者クライアントが接続されている転送サーバの ID が記録されている。

5.2.2 転送サーバ

転送サーバは Web サーバでアップロードされた印刷対象ファイルをプリンタ提供者クライアントへ転送する。どの転送サーバがどのプリンタ提供者クライアントから接続されるかは管理サーバによって決定される。転送サーバは複数台存在しており、各転送サーバの負荷が基準値を超えると台数が自動的に追加され、新規に接続されたプリンタ提供者クライアントは追加された転送サーバに接続される。これによって 1 台の転送サーバに接続が集中することを防ぐ。

5.2.3 プリンタ利用者クライアント

プリンタ利用者クライアントは Web サーバであり、システムを利用するユーザに対して、プリンタ情報の表示やファイルのアップロード等を行う Web インタフェースを提供する。管理サーバから受け取るプリンタ情報の中には、プリンタ提供者クライアントの接続先転送サーバの ID が含まれており、プリンタ利用者クライアントはその ID を元にどの転送サーバにファイルを送信するか決定する。

5.2.4 プリンタ提供者クライアント

プリンタ提供者クライアントは、起動後に起動した PC にインストールされているプリンタドライバの一覧を表示する。提供者は公開するプリンタを選択し、提供者名、プリンタの場所といった情報を入力、公開を行う。公開開始後に、提供者クライアントは管理サーバから転送サーバの状況を取得し、転送サーバの負荷の状況に合わせて接続を

行う。

6. まとめと今後の課題

本節では、本稿の今後の課題とまとめについて述べる。

6.1 今後の課題

今後の課題として、オートスケールの基準項目と閾値の設定、評価項目の設定があげられる。現在、転送サーバのオートスケールを行う基準として、暫定的にCPU使用率が75%を超すと新規転送サーバを起動することでスケールアップを行っている。しかし、先行研究のクライアントサーバ型PrinterSurfではCPUの性能よりも先に通信速度の上限に達することが多かったため、帯域を基準値としたスケールアップ方法の実装と、閾値の設定が必要である。また、先行研究のクライアントサーバ型と性能を比較し、クラウドサーバ型にしたことでどう変化したかを評価するための項目を考察する必要がある。

6.2 まとめ

本稿では、クラウド型プリンタ共有システムの構築に必要な調査と、システムの提案、実装を行った。調査の結果、クラウドサーバは物理サーバではコストのかかるサーバの増設が容易に行え、また自動的にサーバを増減させるオートスケール機能が特徴であることが分かった。そこで、機能を分割した分散型サーバがクラウドサーバに適していると考え、先行研究のクライアントサーバ型PrinterSurfを元に、サーバの機能を分けたクラウド型サーバの実装を行った。今後の展望として、オートスケール機能の基準項目と閾値の考察と、システムの評価が挙げられる。また、管理サーバのスケールアップ機能を実装し、データベースの負荷を分散する機能を実装する必要がある。

参考文献

- 1) 齊藤達郎, 齊藤義仰, 峰野博史, 村山優子: PrinterSurf: モバイル環境に適した印刷システムの設計と実装, マルチメディア, 分散, 協調とモバイル(DICOM02009)シンポジウム論文集, pp. 379-386 (2009).
- 2) 吉田忍, 長嶋呈馬, 齊藤義仰, 村山優子: クライアントサーバ型PrinterSurfシステムの開発, マルチメディア, 分散, 協調とモバイル(DICOM02012)シンポジウム論文集, pp. 1712-1716 (2012)
- 3) R. Herriot, S. Butler, P. Moore, R. Turner and J. Wenn: Internet Printing Protocol/1.1: Encoding and Transport, RFC 2910, 2000
- 4) T. Hastings, R. Herriot, R. deBry, S. Isaacson and P. Powell: Internet Printing Protocol/1.1: Model and Semantics, RFC 2911, 2000
- 5) net print: <http://www.printing.ne.jp/>
- 6) UPnP: <http://www.upnp.org/>
- 7) 株式会社エム・ピー・テクノロジー: <http://www.mpotech.co.jp/>
- 8) PrinterOn: <http://www.printeron.com/>
- 9) 高橋則也, 村山優子: PrinterSurf: ネットワーク上のプリンタを活用するためのシステムの提案, マルチメディア, 分散, 協調とモバイル(DICOM02003)シンポジウム論文集, pp. 709-712 (2003).
- 10) 中上恭介, 後藤幸功, 村山優子: アドホックネットワーク向

け簡易印刷エージェントPrinterSurf, マルチメディア, 分散, 協調とモバイル(DICOM02005)シンポジウム論文集, pp. 581-584 (2005)

- 11) Peter Mell, Timothy Grance: NISTによるクラウドコンピューティングの定義, 独立行政法人情報処理推進機構 (IPA), 2011
- 12) Amazon EC2/S3: <http://aws.amazon.com/jp/ec2/>
- 13) Synergy research group: Cloud Boom Continues as Quarterly IaaS/PaaS Revenues Exceed \$2B (2013).
- 14) Google App Engine: <https://developers.google.com/appengine/>
- 15) Salesforce: <http://www.force.com/jp/>
- 16) Microsoft Windows Azure: <http://www.windowsazure.com/>
- 17) IBM Cloud: <http://www.ibm.com/cloud-computing/>
- 18) Eucalyptus: <http://www.eucalyptus.com/>
- 19) OpenNebula.org: <http://www.opennebula.org/>
- 20) OpenStack: <http://www.openstack.org/>