

携帯機用モバイルサーバミドルウェア

太田 賢 中川 智尋 吉川 貴 竹下理人 倉掛正治

NTT ドコモ マルチメディア研究所

本稿は、厳しいリソース制約や不安定な無線接続、複雑な割り込みを持つ携帯機をサーバ化するため、1. 固定サーバ並の信頼性、2. 移動透過性と位置依存性、3. プライバシとリソース保護を設計目標とした2種類のサーバミドルウェアを提案する。シンサーバミドルウェアがサーバ機能とサーバプログラムをモバイルサーバホストとサロゲートと呼ばれる固定ノードに分割するのに対し、ファットサーバミドルウェアはモバイルサーバホストとクライアントホストの間でエンドエンドに分割する構成をとる。本稿はこの2つのモデルを要求機能の観点と性能の観点から比較する。3つのアプリケーションを例題として、異なるアクセス/更新頻度、データ量、切断パターンを設定したシミュレーションを行い、応答時間と無線トラフィックを調べた。

Mobile Server Middleware for Smart Handset

Ken Ohta Tomohiro Nakagawa Takashi Yoshikawa Masato Takeshita Shoji Kurakake
Multimedia Laboratories, NTT DoCoMo, Inc.

A mobile server on a handset works on an unstable platform with resource constraints, unstable wireless links, and complex interruptions. This paper presents two kinds of middleware for thin and fat handsets to meet our design goal: 1. fixed equivalent reliability, 2. mobility transparency and awareness and 3. privacy and resource protection. The thin server model splits server programs and functions into a mobile server host and a fixed intermediate node called a surrogate, while the fat server model splits them into a mobile server host and a client host in an end-to-end fashion. We compare both models in terms of functionality. We also evaluate the performance including response time and wireless traffic on three types of server applications with different loads and various disconnection patterns by computer simulation.

1 はじめに

Web ブラウザと Java 仮想マシンの搭載により、携帯電話機は電話や E メールなどのコミュニケーションツールから、地図やオンラインショッピング、グループウェア、ゲームなどの多様なネットワークサービスを使うためのツールとなった。現在の携帯機はサーバソケットやサーバプログラム管理、アクセス制御などを備えないため、常にクライアントとして動作している。携帯機のサーバ化は、この通信機能の非対照性を補完するものである。サーバ化により、常時、ユーザの身につけられた携帯機からのリアルタイム情報発信、Peer-to-Peer のパーソナル情報共有など、位置や個人に依存した新たなサービス領域の開発を促進できるものと考えられる。

モバイルサーバには、従来型の WS や PC 上の固定サーバとは異なる機能が要求される。第一に、モバイルサーバは接続性や品質が変動する無線リンクによりネットワーク接続し、着信等による複雑な割り込みが発生する不安定なプラットフォームで動作するため、高信頼なサービス提供を行うための仕組みが必要である。第二に、サーバが動作する携帯機 (モバイルサーバホス

ト,MSH と呼ぶ) はネットワーク的、物理的に移動する可能性がある。MSH が別のネットワークに移動して、IP アドレスが変化したとしても稼働中のセッションは維持可能としなければならない。一方で、物理的位置に基づき、クライアントがモバイルサーバにアクセス可能とするためのサービス発見機構も必要である。第三に、携帯機は極めて個人的な機器であるため、プライバシ保護に加えて、ユーザ自身の携帯機利用のためのバッテリー確保などのリソース保護機構も必要とされる。MIDP2.0[1] や JXME[2] などの既存のモバイルサーバミドルウェアは、これら3つの要求を満たすようには設計されていない。

これらの設計要求に対し、本研究はプロキシ型とエンドエンド型の2つのモバイルサーバミドルウェアを提案する。プロキシ型のシンサーバミドルウェアが、サーバ機能とプログラムを、MSH と固定網上のノード (サロゲートと呼ぶ) に分割するのに対し、エンドエンド型のファットサーバミドルウェアは MSH とクライアントホスト (CH) の間で分割する構成をとる。シンサーバの場合はサロゲート、ファットサーバの場合は CH 上で、サーバプログラムの一部であるサービスプロキシが動作するため、MSH がネットワークから切断状態にあって

も、クライアントはモバイルサーバにアクセス可能となる。両ミドルウェアは、サービスボディ・プロキシ間の同期、障害に対処するセッション維持機構、位置依存サービス広告、動的アクセス制御機構などを備える。

以下、2章でモバイルサーバに特有な上記3つの設計目標を7つの要求機能に分解して説明し、既存のモバイルサーバミドルウェアを概観する。3章では、2つのモバイルサーバミドルウェアを提案し、機能性について比較する。4章では3つのアプリケーションに関して性能を評価し、最後に5章でまとめとする。

2 背景

まず、前提条件としてシン/ファットサーバのホストの通信能力に関する仮定を示す。現在のほとんどの携帯電話はシンサーバに対応し、プライベートIPアドレスを持ち、ゲートウェイを介してインターネットにアクセスする。インターネット上の外部ホストから到達可能なグローバルIPアドレスや、接続を受付けるためのサーバソケットを持たないため、電子メールの受信を除き、他のホストからのメッセージ受信のためには、ポーリングを必要とする。

ファットサーバは、グローバルIPアドレスを持つことができ、外部ホストから、サーバソケットを介して直接メッセージを受信することが可能であるとする。また、Bluetooth や 802.11b のような短距離無線により、物理的な近傍に存在するクライアントにサービスを提供することもできる。現在のほとんどのPDAが、ファットサーバホストの通信能力の仮定を満たす。

2.1 要求機能

固定サーバと同等の安定性 安定性は可用性とロバスト性、スケーラビリティを含む。可用性は、MSHが、電源オフや圏外に位置するなどして、ネットワークから切断状態にあったとしてもサービス提供を維持しなければならないことを示す。ロバスト性は、クライアントやユーザから、突然の無線リンク切断や電源断、着信による割り込みによるセッション障害を隠蔽し、自動的に回復しなければならないことを意味する。そして、スケーラビリティは多数のクライアントからのアクセスを処理し、許容される遅延の範囲で応答することを要求する。

移動透過性と位置依存性 移動透過性はネットワーク的な移動の隠蔽を意味し、MSHが移動先のネットワークで一時的なアドレスを取得することにより、IP アドレ

表 1: 要求機能とミドルウェア

要求機能	MIDP	JXME	Thin	Fat
可用性	低	低	高	中
ロバスト性	低	低	中	高
スケーラビリティ	低	低	高	中
移動透過性	中	中	高	中
位置依存性	低	低	高	高
プライバシー保護	低	低	高	高
リソース保護	低	低	高	高

スに変化しても、クライアントが稼働中のセッションを維持できるようにしなければならないことを示す。位置依存性は物理的に移動して位置依存の情報を発信するMSHのため、その現在位置を反映したサービス発見機構が必要であることを示す。場所の名前や座標(緯度経度)による絶対的な位置指定に加え、相対位置指定(周囲100m以内のサーバ等)もサポートする必要がある。

プライバシーとリソース保護 プライバシ保護は、ユーザの位置や写真など、プライバシーに関わる情報を扱うサーバのため、オーナーのコンテキストに応じたきめ細かなアクセス制御が必要であることを意味する。例えば、ユーザ位置を公開するプレゼンスサーバについて、会社の同僚からのアクセスを就業時間に限って許可したり、周囲の映像を発信するモバイルカメラサーバを、指定範囲(イベント会場内など)にユーザが位置する間だけ公開するなど、モバイル環境特有のアクセス制御が必要になると考えられる。静的なユーザ/グループのリストを保持する従来型のアクセス制御では不十分である。一方、リソース保護は、MSHのオーナー自身の音声通話やメール送受信のためのバッテリーや無線リソースの確保が必要であることを示す。例えば、MSHのバッテリー残量がユーザの指定値30%を下回ったら、サーバへのアクセスを拒否するポリシーが考えられる。

2.2 関連研究

既存のモバイルサーバミドルウェアとしてMIDP2.0とJXMEを紹介し、上記要求機能に対する対応レベルを表1に示す。MIDP2.0(Mobile Information Device Profile 2.0)[1]は、J2ME対応の携帯電話やPDAのためのJavaランタイム環境であり、サーバソケット、プッシュメッセンジング、接続要求に対するアクセス制御、サーバプログラムを起動、終了、中断するためのAMS(Application Management System)等、共通的なサーバ機能が提供さ

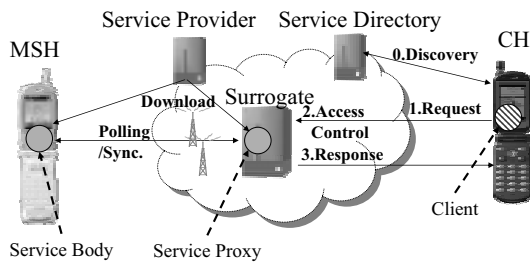


図 1: シンサーバモデル

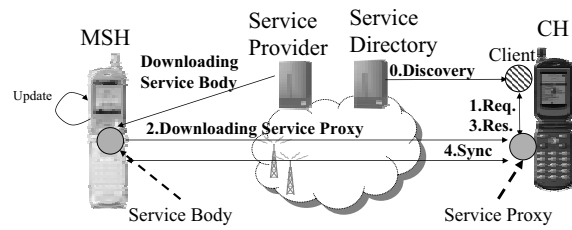


図 2: ファットサーバモデル

3 モバイルサーバミドルウェア

れている。しかし、安定性や保護性、位置依存性に関するサポートは不足している。移動透過性は、動的 DNS 更新 (RFC2136)、や Mobile IP (RFC2002 など) の利用により、サポートすることができる。

一方、JXME[2] は MIDP デバイスが、JXTA のピアツーピアネットワークに参加することを可能にするプロトコルセットである。リソース制約のため、携帯機には最小のプロトコルと機能を実装し、PC や WS 上で実行される JXTA Relay と呼ばれるプロキシが携帯機の代わりに、ピアの発見や XML のパース、携帯機用の軽量バイナリプロトコルと JXTA ネイティブの XML ベースプロトコルの変換を行う。JXME デバイスは NAT の背後に位置することもあるため、JXTA ネットワークから接続を受け付けるためにも、Relay ピアが必要となる。移動透過性は IP アドレスとは独立の JXTA アドレスリング手法により達成されるが、その他の要求機能については満足されない。

本研究のミドルウェアは、既存のモバイルミドルウェア技術を参考にして設計されている。可用性に関して、ROVER[3] は、クライアントに再配置可能なサーバオブジェクトと非同期 RPC 機構により、ネットワークからの切断時も、アプリケーションがブロックせずに動作可能としている。ロバスト性に関して、Pront[4] は、ゲートウェイを利用した移動網用の高信頼非同期メッセージングシステムである。移動透過性について、ALICE[5] はモビリティゲートウェイをクライアントとサーバの間に配置し、クライアントやサーバの移動を吸収する。INS(Intentional Naming System)[6] は位置依存性をサポートするリソース発見・利用システムであり、あるサービスが別の場所に移動する際、サービスの管理を行うネームリゾルバは登録を更新する。

本章は、シンサーバとファットサーバの 2 つのモバイルサーバミドルウェアを提案する (図 1,2)。前提として、サービス提供者はあるサーバアプリケーションに対し、サービスボディとサービスプロキシの 2 種類のプログラムを提供すると仮定し、前者はサーバのオーナーが MSH 上にダウンロードし、後者はシンサーバの場合、サロゲートがサービス提供者からダウンロードするものとする。ファットサーバの場合は CH が MSH からダウンロードを行う。図 3(a) にシンサーバ、(b) にファットサーバミドルウェアのアーキテクチャを示す。ミドルウェアは、サービス広告、セッション管理、同期、サーバ管理、リソース/コンテキスト監視、動的アクセス制御、ポリシー管理の 7 つのモジュールから構成される。

サービス広告モジュールは、モバイルサーバの URL と属性をサービスディレクトリに登録し、クライアントは、ディレクトリを検索して、どのサーバにアクセスするかを決定する。ディレクトリは、UDDI や SLP, LDAP(RFC2251) などの既存のサービス管理技術を利用して構築される。URL に含まれるホスト名とポート番号に関して、ファットサーバの場合は MSH のホスト名とポート番号が登録されるのに対し、シンサーバの場合は、サロゲートのものが登録される。また、登録される属性にはサービスの機能、アクセス制御ポリシー、位置情報などが含まれるが、ユーザのコンテキストの変化に従って位置などが変化した際、本モジュールは再登録を行う。

3.1 シンサーバミドルウェア

シンサーバの場合、サロゲートの動的アクセス制御モジュールが要求を受理するか拒否かを判定した後、サービスプロキシが要求を処理して、応答を返す。サービス提供の間、MSH 上の同期モジュールはあるスケジューリ

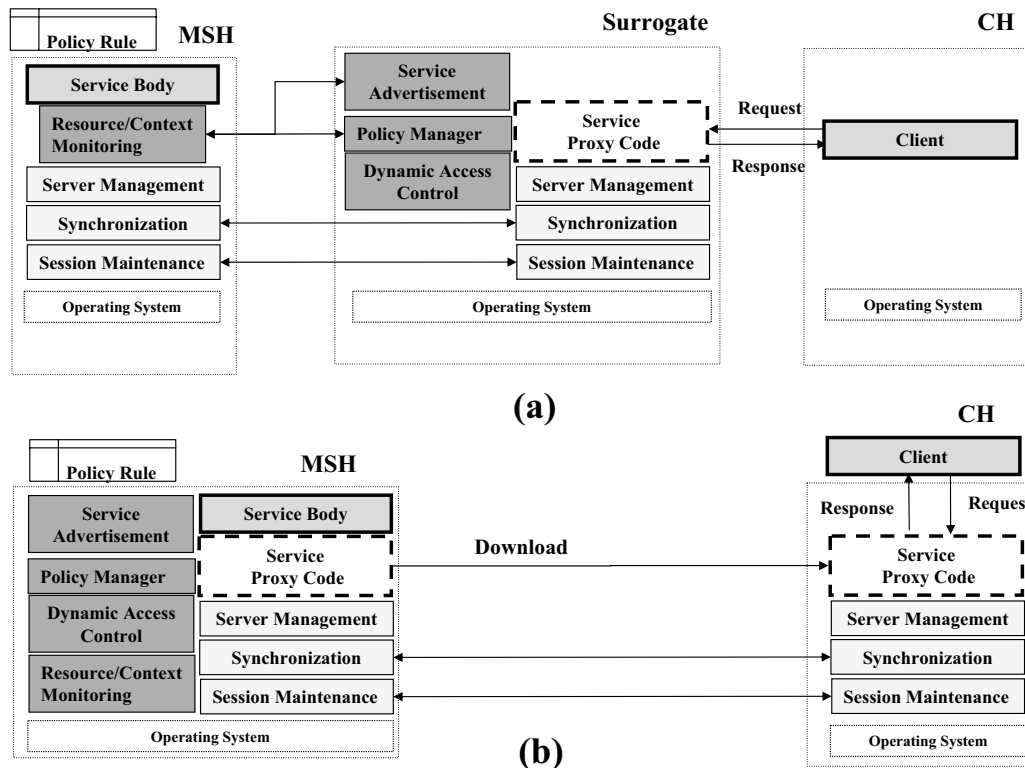


図 3: サーバミドルウェアアーキテクチャ

ング方式に基づき起動して、サービスプロキシにポーリング及び更新通知を行って、サービスプロキシ - ボディ間の状態やデータベースの同期を行う。スケジューリング方式自身の設計は今後の課題であるが、4章のシミュレーションモデルでは無線トラフィック削減のため、サービスボディで状態更新が起きてから 10 秒間待って更新情報を集約し、ポーリング及び更新通知を行うようにしている。

3.2 ファットサーバミドルウェア

ファットサーバの場合、サーバのオーナーがサービスボディを起動し、サービス広告モジュールがサーバをディレクトリに登録することで、準備完了となる(図2)。クライアントが要求を送信すると、CH 上のミドルウェアはそれを横取りして、サーバ管理モジュールを起動し、要求の宛先の MSH からサービスプロキシを CH にダウンロードさせる。このとき、MSH 側の動的アクセス制御モジュールはアクセス可否の判定を行う。ダウンロード後、サービスプロキシは起動され、クライアントの要求を処理して応答を返す。処理の際、サービスプロキシの状態やデータベースが更新される可能性があるが、CH

上の同期モジュールは、その変化をトリガとして起動され、状態更新をサービスボディに通知すると共に、サービスボディ側の更新情報も取得する。4章の評価では上記のような単純な同期方式を利用している。

3.3 ミドルウェアの比較

固定サーバと同等の安定性 両ミドルウェアを機能性の観点で比較する(表1参照)。可用性については、サービスプロキシにより、両モデルのクライアントとも MSH が非接続状態でもサーバを利用可能となる。シンサーバの欠点は、サロゲートの障害がホストされている全モバイルサーバに影響を与えることである。一方、ファットサーバは、クライアントが最初にサービスプロキシをダウンロードする際、MSH が接続状態でなければならないという弱点がある。

ロバスト性については、シンサーバの場合、MSH - サロゲート間のセッションは、セッション管理モジュールのロギングと再送機構により高信頼化され、突然 MSH が切断状態になっても再接続の際にセッションを回復可能である。しかし、CH にはそのような機能モジュールが含まれていないため、サロゲート - CH 間のセッショ

ンは障害を被る可能性がある。一方、ファットサーバの場合、CH - MSH 間のエンドエンドでセッションを高信頼化できる。

スケーラビリティに関しては、サロゲートは一般に携帯機よりも豊富なリソースを持つため、許容レベルの応答時間を維持しながら、より多くのクライアントを収容可能である。ただし、さらに多数のサーバをホストするためには、サロゲートを増やし、負荷分散させる必要がある。一方、ファットサーバの場合、MSH が多数のアクセスを受け付けると、応答時間が悪化する可能性がある。ただし、サービスプロキシが CH 上でローカルに応答を返すことにより、MSH の負荷と無線トラフィックを削減する効果もある。

移動透過性と位置依存性 両者とも、サービスプロキシがクライアントとのセッションを確立するため、MSH の IP アドレスの変化は隠蔽される。セッション管理モジュールは、動的 DNS によって MSH の現在の IP アドレスを解決して再接続し、サービスプロキシ - ボディ間のセッションを回復できる。代替手法として、MSH がモバイル IP をサポートしていれば、再接続の必要なく、セッションを維持できる。ただし、MSH が NAT の背後に移動した場合、セッションが遮断される可能性がある。一方、位置依存性に関しても、サービス広告モジュールがモバイルサーバの位置を随時、ディレクトリに更新登録するため、クライアントは位置依存のシン/ファットサーバを発見可能である。

プライバシー保護とリソース保護 プライバシを保護するため、サロゲート上 (シンサーバ)、あるいは MSH 上 (ファットサーバ) の動的アクセス制御モジュールはアクセス制御リスト (ACL) を維持し、アクセス制限を行う。両者とも、ポリシー管理モジュールが、サーバのオーナーが与えたアクセス制御ポリシーと、コンテキスト/リソース監視モジュールが取得した現在のコンテキスト情報に従って ACL を作成し、更新する。一方、リソース保護に関しては、シンサーバの場合、サロゲートがクライアント要求の処理や悪意のある攻撃のブロックを行うため、MSH のリソース消費は削減される。ファットサーバの場合、MSH 自身がすべての要求の処理とアクセス制御を扱う必要があるため、多数のアクセスにより、無線リンクの占有やバッテリーの枯渇が起きる可能性がある。

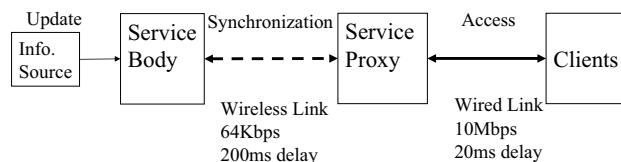


図 4: シンサーバのシミュレーションモデル

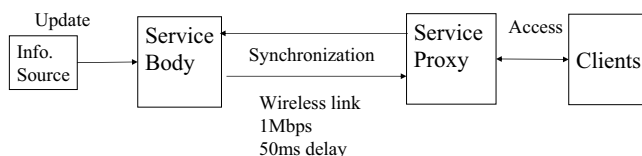


図 5: ファットサーバのシミュレーションモデル

4 性能評価

計算機シミュレーションにより、MSH が送受信する無線トラフィック容量と、クライアントへの応答時間を評価項目として、両モデルウェアを比較する。無線トラフィックは、MSH の貴重な無線帯域幅とバッテリーに直接、影響する重要な指標であり、応答時間は主要な QoS パラメータである。図 4 のシミュレーションモデルにおいて、サービスボディ - プロキシ間は上り下りとも 64Kbps、一方向遅延 200ms の無線リンク、クライアント - サービスプロキシ間は 10Mbps, 20ms の有線リンクであるとする。一方、図 5 のモデルでは、サービスボディ - プロキシ間は 1Mbps, 50ms の無線リンクを利用するが、ローカル監視のシナリオのみ比較のため、シンサーバと同様の 64Kbps, 200ms の無線リンクを利用する。本稿では、無線リンクプロトコルの再送や FEC の機構と独立にシミュレーションを行うため、ビット誤りのないチャネルを仮定している。また、クライアント - プロキシ間は同一ホスト内のため、遅延 0 に設定した。

シミュレーションプログラムは OMNet++[7] を利用して作成し、表 2 に示す 6 つのシナリオに従い、それぞれ 10 時間 (モデル時間) の実行を行った。更新間隔は、図の Info. Source からサービスボディに到着するサーバ状態の更新イベントの到着間隔であり、ポアソン到着モデルに従う。クライアントからの要求もポアソン到着モデルに従って、以降の図の Access Rate (単位時間の要求到着数) でサービスプロキシに到着する。更新・要求・応答は各メッセージのサイズを示す。サービス時間は、要求がサービスプロキシで処理されるのにかかる時

表 2: シミュレーションパラメータ

シナリオ	更新間隔 (s)	データサイズ (KB)			サービス時間 (s)		同期時間 (s)	
		更新	要求	応答	シン	ファット	シン	ファット
ローカル監視 1	100	100	0.2	100	0.01	0.01	0.01	0.1
ローカル監視 2	5	0.5	0.2	0.5	0.01	0.01	0.01	0.1
地理的チャット 1	200	30	30	300	0.02	0.02	0.01	0.2
地理的チャット 2	50	1	1	10	0.02	0.02	0.01	0.2
パーソナル Web1	1000	100	0.2	500	0.01	0.01	0.01	0.05
パーソナル Web2	100	10	0.5	100	0.01	0.01	0.01	0.05

間で指数分布に従う。

- ローカル監視サーバ: センサから取得した画像や音、温度などのローカルの監視情報を発信する。クライアントはリードオンリーであり、サービスボディのみがサービスプロキシの状態を更新する。
- 地理的チャットサーバ: センサから取得した画像や音、温度などのローカルの監視情報を発信する。クライアントはリードオンリーであり、サービスボディのみがサービスプロキシの状態を更新する。
- パーソナルウェブサーバ: Blog や写真アルバム、個人オークションなど、パーソナルなコンテンツを提供する。このシナリオではクライアント - サービスプロキシ間で、要求 / 応答メッセージが 3 回、交換されるトランザクション的セッションを扱う。

一方、同期時間は、シンサーバの場合、サービスプロキシ（ファットサーバの場合はサービスボディ）がサービスボディ（ファットサーバの場合はサービスプロキシ）から同期要求を受け取って応答を返す際の処理時間である。ローカル監視 1,2 と地理的チャット 1,2 のシナリオでは、CH はすでにサービスプロキシをダウンロード済みであると仮定し、サービスプロキシは要求を受け取ると、サービスボディに同期要求を送って最新情報を獲得し、応答を返す。一方、パーソナル Web サーバ 1,2 の場合、セッションの最初にそれぞれ、250KB と 2KB のコードサイズを持つサービスプロキシを CH にダウンロードした後、サービスプロキシが 3 つの要求に対して 3 つの応答を返し、サービスボディに更新状態を通知してトランザクションを完了するものとする。

4.1 応答時間

図 6,7, 8 に、異なるアクセス率における各シナリオの平均応答時間を示す。シンサーバの場合、サロゲートの

豊富な CPU パワーと帯域幅を利用できるため、ファットサーバに比べて応答時間を短くでき、アクセス増加にも耐える。ここでは、サロゲート上に 1 つのサーバのみがホストされている状況を仮定したが、アクセス率を 0.1 に設定した際の、サーバ数の応答時間に対する影響を図 10 に示す。この結果から、ある許容応答時間内で、サロゲート上にいくつのサーバを収容できるかを見積もることができる。サロゲートが収容可能なチャットサーバと監視サーバの数は異なるが、これはチャットがより多くのリソースを消費するためである。応答性を維持するためには、複数のサロゲートを用意し、サーバ種別によるリソース消費の差を考慮して負荷分散を行う必要があることが分かった。

一方、ファットサーバの場合、アクセス率の増加と共に、応答時間は急激に増加する。軽量データを扱う Fat2 の場合は多少、増加が抑制されるが、いずれにしても、ファットサーバにはスケラビリティに関して大きな問題があることが分かった。特に無線リンクの帯域幅がボトルネックとなっている。ただし、例外として、パーソナル Web の Fat2 はシンサーバよりも短い応答時間を達成している。これは、ダウンロードするサービスプロキシのサイズが小さく、要求と応答メッセージが無線リンクを横切らないためである。大容量のプロキシをダウンロードする Fat1 は大きな遅延を被っており、プロキシのコードが小さいか、CH にキャッシュされている場合、トランザクション型サーバにはファットサーバモデルが有効であることが分かった。

さらに、MSH の切断の応答時間への影響を調べるため、パーソナル Web の Fat1 シナリオでアクセス率を 0.1 に設定し、5 つの切断パターンでシミュレーションを行った（図 9）。シンサーバの場合、サロゲートによって MSH の切断は隠蔽されるため、影響はない。1 秒の切断（図の D1）はハンドオフ、5 秒と 10 秒の切断（D5, D10）はバースト誤り、30 秒と 60 秒（D30, D60）は MSH の一時的なダウン（電源オフや圏外）を意図している。デー

タ転送は再接続の際、即座に再開されるものと仮定しており、ここでは単純のため、特定のトランスポート/セッションプロトコルのオーバーヘッドを考慮していない。図より、MSHの一時的なダウン時間が長くなるほど、応答時間に大きな影響を与えることが確認された。ファットサーバモデルで安定的なサービス提供を行うには、サービスプロキシや状態を固定ノードにキャッシュさせるなど、可用性を向上させる仕組みが必要であることが分かった。

4.2 無線トラフィック容量

図 11,12 に、サービスボディ - プロキシ間の無線トラフィック量を示す。ローカル監視シンサーバの場合、MSH はセンサデータを発信するのみで、クライアントからの要求の処理を行わないため、アクセス率が増加してもトラフィックは一定に抑えられる。一方、チャットのシンサーバの場合、クライアント要求の処理がサービスプロキシの状態更新を引き起こすため、アクセス率の増加と共に、サービスボディとの同期トラフィックが大きく増加する。効率的な同期スケジューリング、プロトコルが必要であることが分かった。

一方、ファットサーバの場合、アクセス率が増加するとトラフィックが急激に増加し、無線帯域幅とバッテリーを多く消費することが分かった。ただし、オンデマンドに動作するファットサーバは要求が到着しない限り、トラフィックを生成しないため、アクセス率が低い場合はシンサーバよりも消費トラフィックを削減可能である。固定ノード上へのキャッシュなど、MSH へ到着するアクセス数を減らす仕組みが必要であることが分かった。

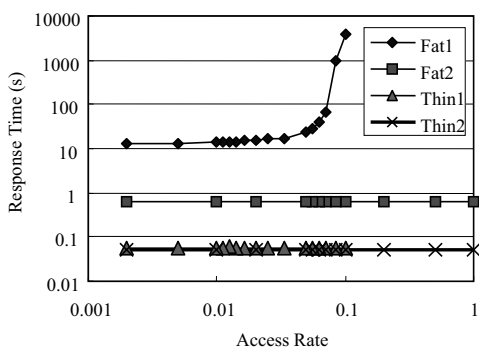


図 6: ローカル監視サーバの応答時間

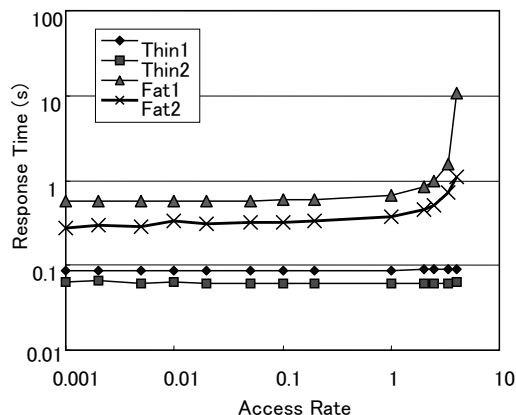


図 7: 地理的チャットの応答時間

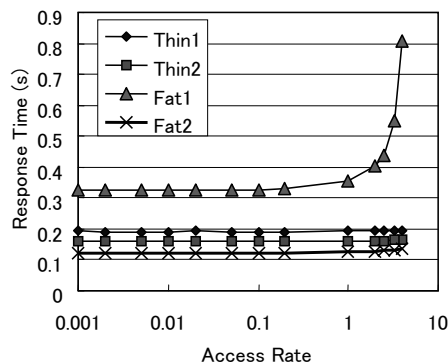


図 8: パーソナルウェブサーバの応答時間

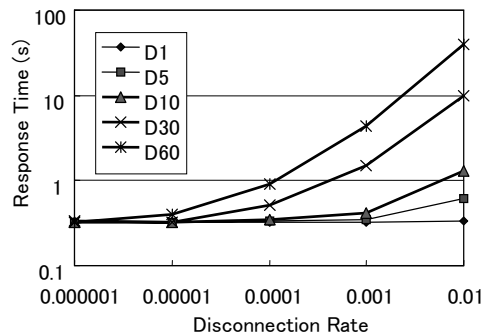


図 9: 切断の応答時間に与える影響

5 まとめ

本稿では、携帯機用のモバイルサーバ実現のために要求される機能性を整理し、安定性と移動性と保護性の3つを設計目標として、プロキシ型とエンドエンド型の2つのミドルウェアを検討し、評価した。その結果、両者とも設計目標を満たすが、プロキシ型がほとんどの場合で応答時間と無線トラフィック量に関して有利に動作することが分かった。ただし、アクセス率が低く、更新レートが高いサービスの場合や、トランザクション型のサーバの場合、ファットサーバモデルのほうが有効になる場合がある。今後の課題としては、効率的な同期スケジューリングアルゴリズムの設計、サーバアプリケーションの開発を支援するためのモバイルサーバAPIの開発、実際の無線リンクプロトコル、トランスポートプロトコルを利用した環境での詳細なシミュレーション、両モデルの実装などが挙げられる。

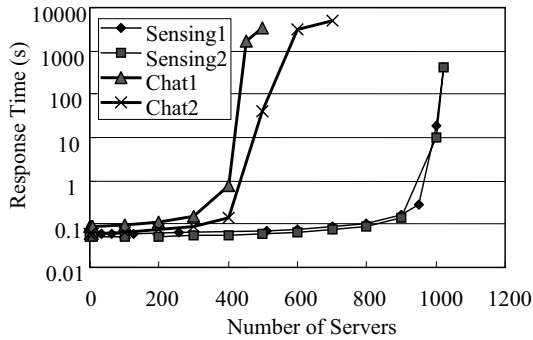


図 10: サーバ数に関するスケーラビリティ

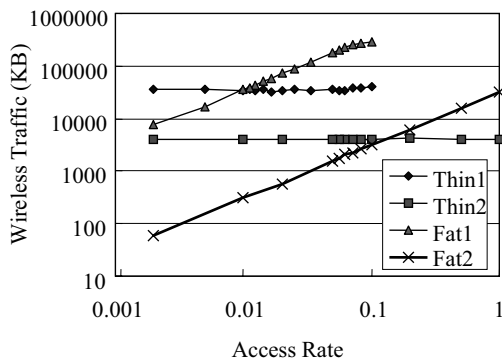


図 11: ローカル監視サーバの無線トラフィック

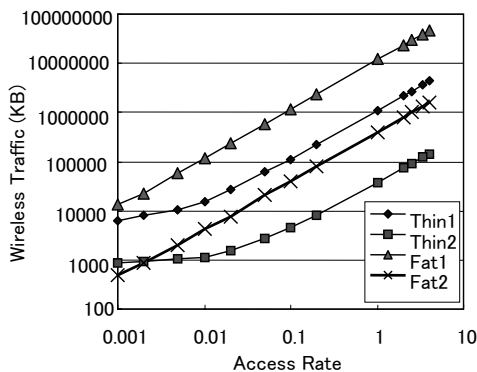


図 12: 地理的チャットの無線トラフィック

参考文献

- [1] Mobile information device profile, v2.0 (jsr-118), 2002. <http://jcp.org/aboutJava/communityprocess/final/jsr118>.
- [2] C. W. Akhil Arora and K. S. Pabla. Jxta j2me implementation project, 2003. <http://jxme.jxta.org>.
- [3] Anthony D. Joseph, Joshua A. Tauber, and M. Frans Kaashoek. Mobile computing with the rover toolkit. *IEEE Transactions on Computers*, Vol. 46, No. 3, pp. 337–352, 1997.
- [4] Eiko Yoneki and Jean Bacon. Gateway: A message hub with store-and-forward messaging in mobile networks. In *23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*, pp. 348–353, May 2003.
- [5] Mads Haahr, Raymond Cunningham, and Vinny Cahill. Towards a generic architecture for mobile object-oriented applications. In *SerP 2000: Workshop on Service Portability*, December 2000.
- [6] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. The design and implementation of an intentional naming system. In *Symposium on Operating Systems Principles*, pp. 186–201, 1999.
- [7] Andras Varga. The omnet++ discrete event simulation system. In *The European Simulation Multiconference (ESM'2001)*, 2001.