

## 携帯機用 Grid ミドルウェアの設計と実装

太田 賢<sup>†</sup> 吉川 貴<sup>†</sup>  
中川 智尋<sup>†</sup> 稲村 浩<sup>†</sup>

本論文は、サーバとして動作する携帯機のリソース制約と不安定性を隠蔽するため、適応的 QoS フレームワークに基づくモバイル Grid ミドルウェアを提案する。モバイル Grid システムにおいて、携帯機上のコード、データは固定網の Grid プロキシに複製され、可用性を向上させる。しかし、複製の新鮮さや応答性の維持のための同期やセッション管理機構がバッテリー消費を増加させ、稼働時間（持続性）を制限する。本フレームワークは、携帯機と固定網上の Grid プロキシ間における同期スケジューリングとセッション管理において、アプリケーションごとに異なる可用性、応答性、持続性の QoS 要求に基づき、同期の集約時間とポーリング間隔を最適化する。さらに、残存バッテリー容量の減少や接続性の悪化等のリソース状態の変化に適応するため、目標 QoS の低減による持続時間の拡大、集約時間の短縮化による可用性の維持をはかる。本フレームワークに基づく Grid ミドルウェアを現行の J2ME ベースの携帯機に実装し、Grid プロキシを Globus Toolkit を配備した PC 上へ実装した。同期時間、応答時間、持続時間を測定した結果、提案方式がアプリケーションごとに異なる可用性・応答性を維持しつつ、12～35%の稼働時間の拡大を達成することを確認した。

## Design and Implementation of Mobile Grid System for Handsets

KEN OHTA,<sup>†</sup> TAKASHI YOSHIKAWA,<sup>†</sup> TOMOHIRO NAKAGAWA<sup>†</sup>  
and HIROSHI INAMURA<sup>†</sup>

This paper proposes mobile Grid middleware based on an adaptive QoS framework to conceal the instability and resource constraints of mobile server hosts (MSHs). In a mobile Grid system, data and codes on an MSH are replicated on a fixed Grid Proxy for enhanced availability. However, sustainability (operation time) is limited by battery consumption due to the synchronization needed to maintain the freshness of replicas and the session management needed to maintain responsiveness. According to application QoS requests, the framework optimizes synchronization scheduling and polling intervals to maximize sustainability. It also adapts to dynamic resource changes including low residual battery and poor wireless connectivity, by decreasing the target QoS and by reducing the aggregation period for synchronization. We built a prototype of mobile Grid middleware on a handset and Grid Proxy on PC running Globus Toolkit 3, and measured synchronization time, response time, and operation time. Experiments showed that the framework enables MSH to increase its operation time by 12 – 35% while meeting diverse application QoS request.

### 1. はじめに

Grid は、異種システムにまたがって分散された計算機やデバイス、データ、情報、知識等のリソースを共有するためのプラットフォームである。異種システムの相互運用性のため、Global Grid Forum による標準化や基盤ソフトウェアの共通化が進んでおり、Globus Tool Kit<sup>1)</sup> は、デファクトの Grid ミドルウェアとして、シングルサインオンや遠隔ジョブ実行、ファイル転送、リソース管理等の機能を提供している。

従来の Grid システムは主に固定網上の PC やワークステーションから構成されていたが、Legion<sup>2)</sup>、LEECH<sup>3)</sup> 等のプロジェクトは、携帯電話機（携帯機と呼ぶ）や PDA、ラップトップ PC を含むモバイル Grid システムを構築している。モバイルノードには、Web ブラウザ等を介した Grid へのインタフェース<sup>4)</sup> だけでなく、データや処理リソースを Grid 上のクライアントに提供するサーバとしての役割が与えられる。現在の携帯機は、カメラや GPS 等のセンサやプログラムの実行環境を内蔵しており、携帯機をウェアラブルな情報発信サーバとして Grid に組み込むこ

<sup>†</sup> NTT ドコモマルチメディア研究所  
Multimedia Laboratories, NTT DoCoMo, Inc.

<http://www.gridforum.org/>

とで、ブックマークやアドレス帳、音楽データ等を交換する共有系アプリケーション、周囲の画像や音声、位置、体温、脈拍等の個人の時系列情報をアーカイブするログ系アプリケーション等、新たなサービスが実現可能であると考えられる。

本研究は、サーバ化された携帯機 (Mobile Server Host; MSH と呼ぶ) のリソース制約と不安定性を、固定網上の Grid プロキシによって隠蔽するアプローチでモバイル Grid システムを構築する。MSH は無線接続かつバッテリー駆動のデバイスであるため、通信圏外やバッテリー枯渇の状態にある場合、クライアントからの MSH 上のデータやコード (ローカルオブジェクトと呼ぶ) へのアクセスは失敗する。ローカルオブジェクトを Grid プロキシに複製し、更新があった際に指定のデッドライン以内に同期することで、その可用性と新鮮さを維持できる。また、サーバ socket をサポートしていない MSH や、NAT (Network Address Translation) を行うゲートウェイの背後に位置する MSH の場合、MSH がクライアントからのアクセスを直接、受け付けできない制約がある。Grid プロキシが代理でアクセスを受け付け、MSH が指定のデッドライン以内にポーリングを行うセッション管理機構によって、応答性を維持できる。

しかし、従来のモバイル Grid システムは、可用性・応答性と MSH の持続性とのトレードオフの解決や、無線接続品質やバッテリー残量等の動的なリソース状態の変化に対する適応能力を持たないため、MSH の稼働時間が制限される可能性がある。可用性維持のための同期、応答性維持のためのセッション管理によるバッテリー消費は、稼働時間を制限する。一方で、バッテリー消費の節約のために更新を集約して同期を遅らせる場合、MSH の不意の無線リンク切断によって、同期がデッドラインの前に完了せず、可用性が低下する可能性がある。また、アプリケーションやユーザが稼働時間を優先する QoS 要求を持つ場合、バッテリー残量が少ない状態に適応した動作が必要となる。

本論文は、アプリケーションの多様な QoS 要求と動的なリソース状態の変化に適応可能とするための適応 QoS フレームワークを提案し、携帯機用 Grid ミドルウェア (Mobile Server Middleware; MSM と呼ぶ) に組み込むものである。本フレームワークは QoS ポリシ管理部と端末リソース監視部、スケジューラ部の 3 つのモジュールから構成される。第 1 に、QoS ポリシ管理部はアプリケーションから同期時間と応答時間のデッドライン要求と目標の QoS 保証率を入力する。第 2 に、端末リソース監視部は、周期的に端末の残存

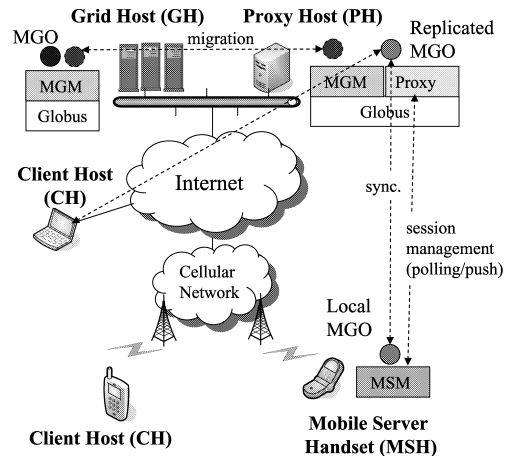


図 1 モバイル Grid システム

Fig. 1 Basic architecture of Mobile Grid System.

バッテリー容量や無線リンクの平均の切断率、切断時間を測定する。第 3 に、スケジューラ部は、目標の QoS 保証率でデッドライン要求を満足するように、同期の集約時間とポーリング間隔を調節する。残存バッテリー容量が少ない場合、バッテリー節約モードに移行し、目標 QoS 保証率を下げることで持続性を維持する。

現行の J2ME 携帯機上に MSM, Globus Toolkit を配備した PC 上に Grid Proxy のプロトタイプを実装した。テストアプリケーションとして、複数のユーザで Web 上のサービス知識を交換するブックマーク共有と、画像を自動で撮影、アップロードするフォトログの 2 つを実装した。セルラ網とインターネットを介した実環境において、MSH と Grid プロキシ間の同期時間と応答時間、持続時間を測定した結果、スケジューラが同期・応答のデッドライン要求を満たしつつ、MSH の持続時間を拡大できることを確認した。

以下、2 章においてモバイル Grid システムの基本構成について述べ、関連研究を概観する。3 章では、適応 QoS フレームワークの各モジュールの機構を説明する。4 章と 5 章ではプロトタイプの実装とその評価について述べ、最後に 6 章でまとめとする。

## 2. モバイル Grid システム

### 2.1 基本アーキテクチャ

モバイル Grid システムは、サーバ化された携帯機 (MSH)、固定網上の Grid プロキシ (PH) とその他の Grid ホスト (GH) の 3 種類の Grid ノードから構成される (図 1)。クライアントホスト (CH) は、MSH にアクセス要求を発行する Grid ノードである。本システムは、携帯機上のデータとコードを、移送・複製可能なモバイル Grid オブジェクト (MGO) にカプ

セル化して扱う。MGO がモバイル Grid アプリケーションを形成する。たとえばブックマーク共有アプリケーションは、コードとしてブックマーク管理プログラム、データとして Web ページのタイトルや URL を含む MGO によって構成される。

MSH, PH, GH の各 Grid ノードに以下の 1. モバイルサーバミドルウェア (MSM), 2. Grid プロキシ, 3. モバイル Grid ミドルウェア (MGM) が配備される。第 1 に, MSM は PH にローカルの MGO の複製を配置し, 同期を維持する機能と, ポーリングやプッシュによって Grid プロキシから CH の要求を受け取るセッション管理機能を備える。第 2 に, Grid プロキシは MGO の複製を作成し, 同期を維持する機能, MSH の代わりに CH からの要求を受け付け, プッシュやポーリングによって MSH に転送するリレー機能, 不正アクセスや DoS (サービス拒否) 攻撃から MSH のリソースを保護するアクセス制御機能, MSH 提供のサービスを LDAP 等のディレクトリに登録する広告機能を提供する。第 3 に, MGM は MGO の論理名と実アドレス (URL 等) のバインド情報をデータベースで管理し, MGO の位置にかかわらず, 名前による参照を解決するネーミングサービスを提供する。これにより, Grid ノードは, MGO があるノードから別のノードへ移動したとしても一貫した論理名によってアクセス可能となる。MGM は, Globus Toolkit のサービスを利用して MGO を管理する。

## 2.2 関連研究

モバイルノード向けのリソース共有ミドルウェアとして, モバイル Grid システム Legion<sup>2)</sup>, 携帯機用のピアツーピアミドルウェア JXME (JXTA for J2ME)<sup>5)</sup>, 耐切断ミドルウェア ROVER<sup>6)</sup> という 3 種のシステムを概観する。これらは携帯機のリソース制約への対処のため, プロキシによる機能分担や再構成可能な設計をとり, 不安定性の隠蔽のために移送や複製, 同期の機能を提供している。しかし, いずれも QoS のトレードオフを解決する能力や, 無線の接続品質, バッテリ残量の変化に適應する機構は実現されていない。

Legion<sup>2)</sup> はシステムコンポーネントを入れ替え可能な設計により, メモリ制約や帯域幅制約が多様なモバイル環境に適應可能とする。また, オブジェクトの移送や複製に追従して一貫した名前によるアクセスを可能とする。リンク切断後に再接続した際, 名前とオブジェクトのバインドを更新する機能も持つ。

JXME (JXTA for J2ME)<sup>5)</sup> は MIDP/CLDC 対応のデバイスが JXTA 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。リソース制約から, 携

帯機に実装するプロトコルと機能を最小化し, PC 等の上で動作するリレーピアが JXME の代わりに, リソース発見や XML パース, 軽量バイナリプロトコルと XML ベースプロトコルの変換を行う。NAT の裏に位置する JXME デバイスは要求メッセージ受信のため, 固定間隔 (デフォルトで 20 秒) でリレーピアにポーリングを行う。

最後に ROVER は, クライアントに移送可能なサーバオブジェクトをインポートすることで, サーバとの接続が切断されていてもクライアントが RPC を発行することを許す。オブジェクトに対する RPC のログを記録し (Queued RPC 機構), 再接続時にオリジナルのサーバオブジェクトに反映する。同期における衝突の解決はアプリケーションが責任を持つ。

その他, 同期スケジューリングについて, リアルタイム Web アプリケーションのデッドライン要求を満足しながら, 同期トラフィックを最小化するアルゴリズムが提案されている<sup>7)</sup>。バッテリー問題について, Odyssey ミドルウェアは, エネルギーの需要と供給を監視してアプリケーションに通知し, ユーザが指定した稼働時間を満足するようにアプリケーションレベルの適應制御 (画面サイズの削減等) を行うことを可能とする。最後に, PARM<sup>8)</sup> はキャッシングや位置管理, 暗号化等のミドルウェアコンポーネントを入れ替え可能にして, 固定ノード上のプロキシと分散処理することで, 適應的なバッテリー節約を可能にする。

## 3. 適応 QoS フレームワーク

本フレームワークの目的は, MSM の同期とセッション管理において, アプリケーションの QoS 要求のトレードオフを解決すること, 動的な端末リソース状態に適應することである。図 2 に MSH のアーキテクチャを示す。本フレームワークは MSM に組み込まれる。QoS ポリシ管理部はアプリケーション (MGO) から同期時間と応答時間に関するデッドライン要求を入力し, 端末リソース監視部は端末のバッテリー残量と無線リンクの接続品質の測定を行う。スケジューラ部はデッドライン要求を満たしつつ, 持続時間を最大化するため, そして切断率やバッテリー残量の変化に適應するため, 同期の集約時間とポーリング間隔を調節する。

### 3.1 QoS ポリシ管理部

本管理部は可用性と応答性の要求として, アプリケーションの同期時間と応答時間のデッドラインを, パラメータの組 (A, R) として入力する。

- 同期デッドライン A: MSH 上のローカルオブジェ

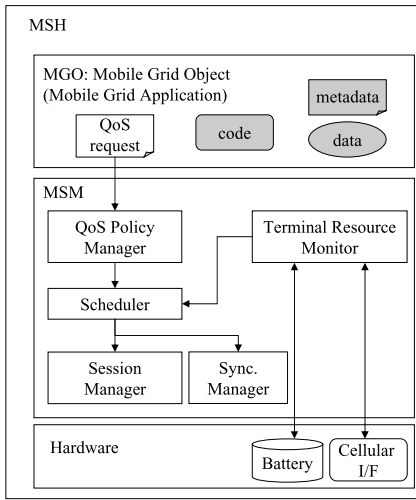


図 2 携帯機サーバのアーキテクチャ  
Fig. 2 Mobile server handset architecture.

クトの更新が、複製と同期されるまでの経過時間  
の上限である。たとえば、フォトログアプリケー  
ションが新規に撮影した画像を複製としてアップ  
ロードするまでのデッドラインを 300 秒と要求  
する。

- 応答デッドライン  $R$ : 複製からの更新要求や同期  
要求を、MSH が受け取るまでの経過時間の上  
限である。たとえば、ブックマーク共有アプリケー  
ションにおける複製の更新を、MSH に通知する  
までのデッドラインを 90 秒と指定する。もう 1  
つの例として、フォトログアプリケーションにお  
いて、CH が最新画像を要求した際、同期要求を  
MSH に通知するデッドラインを 60 秒と指定する。

また、ユーザやアプリケーションの持続性に関す  
る要求として、バッテリー節約モードに移行するバッテ  
リ残量の範囲  $(H, L)$  と、通常時と節約時の目標 QoS  
保証率  $(P, Q)$  も入力する。たとえば、バッテリーのフル  
充電の際の容量を 1 として  $(H = 0.5, L = 0.2)$  ,  
 $(P = 0.95, Q = 0.8)$  と指定することにより、バッテ  
リ残量が 0.2 から 0.5 の場合にバッテリー節約モードに  
移行し、目標 QoS 保証率を 0.95( $P$ ) から 0.8( $Q$ ) に落  
とすというポリシーを設定できる。バッテリー残量が 0.2  
以下であった場合、MSH のオーナーの利用のために、  
サーバ動作を停止する。

3.2 端末リソース監視部

本監視部は、周期的に端末のバッテリー残量  $b$  と無線  
リンクの切断率  $c$  , 平均切断時間  $d$  を測定する。また、  
バッテリー残量  $b$  に従って、MSH の動作モードを  
通常モードかバッテリー節約モードに設定する。 $c, d$  は

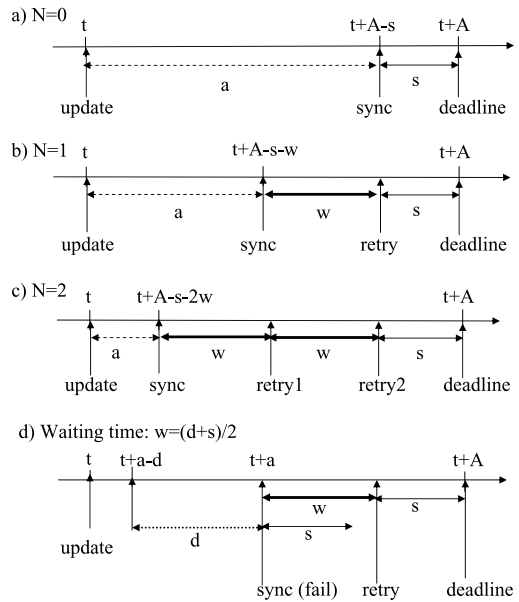


図 3 同期スケジューリング  
Fig. 3 Synchronization scheduling.

無線インタフェースの利用可能性 ( 圏内か圏外 ) のロ  
グをとり、それぞれ過去  $TWIN$  時間の切断イベント  
が起きた回数と、平均の切断時間を計算する。タイム  
ウィンドウ  $TWIN$  は管理者が設定するパラメータで  
あり、長く設定すると MSH のストレージを圧迫する  
が、短い設定は無線接続品質の見積りの精度を低下さ  
せる可能性がある。

3.3 スケジューラ部

スケジューラ部は、目標 QoS 保証率  $p$  , ポーリング  
間隔  $i$  , 同期処理前の集約時間  $a$  の 3 つのパラメータ  
を決定する。 $p$  は通常モードの場合  $P$  , 節約モードの  
場合は  $Q$  に設定する。 $i$  は、応答デッドライン  $R$  と  
 $p$  から式  $i = R/p$  で計算する。通常モード ( $p = P$ )  
に比べ、節約モード ( $p = Q$ ) は間隔が長く設定され  
ることになる。

集約時間  $a$  は、通信データ量の削減のため、MSM  
上のローカルオブジェクトが更新された後、MSM が  
Grid プロキシとの同期処理を実行するまでの待ち時  
間であり、同期デッドライン  $A$  と平均同期処理時間  
 $s$  , 見積りのリトライ回数  $N$  から式 (1) で計算する。  
 $s$  は同期データのサイズや端末の CPU , 無線リンク速  
度に依存するため、測定から得る必要があるパラメ  
ータである ( 5.1 節で述べる ) 。

$$a = A - s - \frac{N(d + s)}{2} \tag{1}$$

図 3 の (a) , (b) , (c) に示すように、 $a$  は接続性が  
安定している環境で最大化できるが ( $N = 0$ ) , 切断

により同期が失敗する確率が高い場合 ( $N = 1, 2$ ), リトライの待ち時間  $w$  とリトライ回数  $N$  を考慮して短縮する必要がある. 図の (d) は, MSM の同期処理 (図の sync(fail)) が切断により失敗し, 同期開始時刻から待ち時間  $w$  後にリトライ (図の retry) を開始する際の, 待ち時間  $w$  の計算例を示す. 本論文は, 待ち時間  $w$  を同期処理の開始時刻から, リトライの開始時刻までの時間間隔であると定義する.

リトライは (1) MSH でのデータ更新後 (時刻  $t$ ), 集約時間が経過した際に同期処理を始めてから (時刻  $t + a$ ), 同期処理時間  $s$  が経過する前に切断が発生した場合と (2) MSH でのデータ更新後, 集約時間が経過した際に同期処理を始めようとした際, 接続が切断状態にあり, 再接続まで待たざるをえない場合の 2 通りで発生する. 図 (d) の同期開始時刻から平均切断時間前の  $t + a - d$  以降から同期処理の完了時刻の  $t + a + s$  までに切断が起きた場合, 同期が失敗する. この 1 回のリトライの場合,  $w$  の平均は  $(d + s)/2$  と見積もることができる. 本論文ではリソース制約のある携帯機上での計算処理削減のため, この  $w$  の値をベースとして,  $N$  回のリトライの場合の待ち時間をこの  $N$  倍として近似的に見積もるようにした (式 (1)). この近似が与える同期デッドライン保証率への影響についての解析は今後の課題である.

一方, リトライ回数  $N$  は, 同期処理が切断によって失敗する確率  $f$  から見積もることができる. 1 回目のリトライ発生の確率を見積もる際, 切断は平均  $d$  時間続いたため, 同期処理時間  $s$  だけでなく, 同期開始から  $d$  時間前からの  $(d + s)$  の区間の切断確率を考慮する必要がある. そこで  $f$  を  $d + s$  時間に少なくとも 1 回切断が起きる時間として, 切断のイベントがポアソン分布で到着するものと仮定すると, 式 (2) で計算できる.

$$f = 1 - e^{-c(d+s)} \tag{2}$$

なお, 2 回目以降のリトライは  $s$  の区間における切断確率を考えるべきであるが, 計算量削減を目的に前記の  $f$  の値を再利用することとした. 切断確率を実際より高く見積もることになるが, この近似の影響の解析も今後の課題である. そして, 成功確率  $1 - f$  と目標 QoS 保証率  $p$  を比較し,  $N$  を計算する.  $p \leq 1 - f$  であれば 1 回の同期で  $p$  以上の確率で同期が完了すると見積もることができ,  $N=0$  となる. 同様に,  $f^2 \leq 1 - p < f$  の場合は  $N = 1, f^2 > 1 - p \geq f^3$  の場合,  $N = 2$  となる.  $N \geq 3$  の場合も同様の手順で計算される.

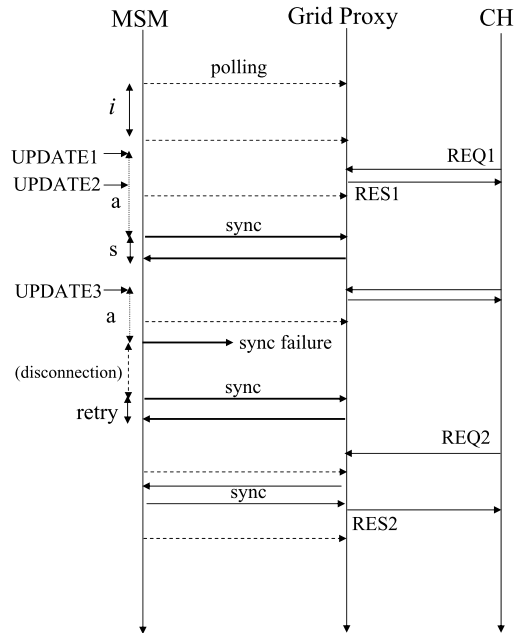


図 4 セッションプロトコル  
Fig. 4 Session protocol.

### 3.4 セッションプロトコル

図 4 に, MSM と Grid プロキシ間のセッションプロトコルを示す. MGO を最初に起動する際, MSM は Grid プロキシ上に複製を作成する. そして周期的にスケジューラを呼び出し, 目標 QoS 保証率, 集約時間, ポーリング間隔を決定する.

MSM はポーリング間隔  $i$  で Grid プロキシへポーリングを行う. MSM が Grid プロキシからプッシュ通知を受信可能な場合, 周期的なポーリングを行う必要はない. ただし, MSH が切断やパワーオフから回復してネットワークに再接続する際, 切断中に保留されたプッシュ通知を受信するためのポーリングを行う. MSM はローカルオブジェクトが更新された際, 集約時間  $a$  後に Grid プロキシと同期を行う. 図の UPDATE1, 2 のように, 集約時間中に起きた更新をまとめることで, トラフィックや消費電力を削減する. 一方, 切断によって同期処理が失敗した場合 (UPDATE3), 再接続時にリトライする.

一方, Grid プロキシ上の複製は, CH から要求メッセージを受け取り, 応答を返す (図の REQ1, RES1). 複製側の更新や同期要求があった際, Grid プロキシはポーリングやプッシュのメカニズムで同期要求を MSM に渡す. 図において, Grid プロキシは REQ2 を受信後, 同期要求を MSM に発行し, 同期の完了後, CH に応答 (RES2) を返している.

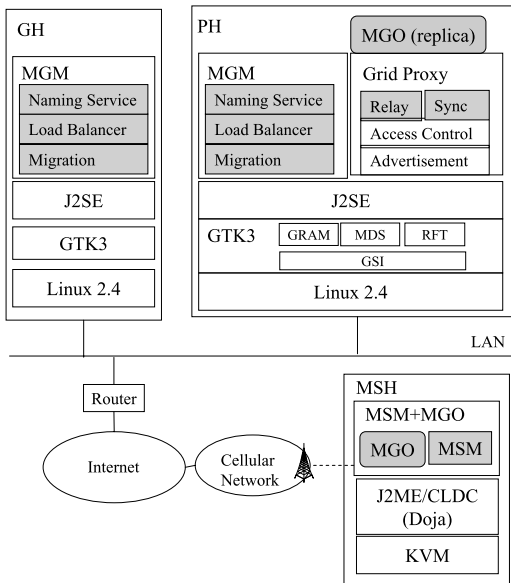


図 5 プロトタイプシステム  
Fig. 5 Prototype system.

#### 4. 実 装

適応 QoS フレームワークに基づき、現行の第 2, 3 世代 (2G, 3G と呼ぶ) の J2ME ベースの携帯機に MSM を実装し、Globus Toolkit を配備した PC 上へ Grid プロキシと MGM を実装した。図 5 に、プロトタイプのネットワーク構成を示す。図の灰色のコンポーネントが新規に開発した部分に相当する。

##### 4.1 携帯機の制約

現行の J2ME/CLDC ベースの携帯機は、同時に複数の Java アプリケーションを起動する能力やクラスローダがないという機能的制約と、アプリケーションサイズ (JAR ファイル) やデータストレージ (ScratchPad と呼ばれる) の容量制限がある。プロトタイプで利用した 2G, 3G 携帯機の場合、それぞれ JAR ファイルサイズは 30 KB, 100 KB, ScratchPad は 200 KB, 400 KB に制約される。そこで、MGO のコード部と MSM を 1 つのアプリケーションとして実装することとした。また、MGO のデータ部について変数や状態等の小容量のメタデータと、画像等の比較的大容量のバイナリデータを分け、前者はメモリ中に、後者はスクラッチパッドに保持するようにしている。

プロトタイプにおいて MSM 単独での JAR サイズは 19.7 KB であり、そのうち XML パーサである ASXMLP が 5.6 KB を占める。メタデータや MGO

表 1 モバイル Grid アプリケーション  
Table 1 Mobile Grid application.

アプリケーション	ブックマーク	フォトログ	
データ: サイズ, レート	メタデータ (KB) バイナリ (KB) 平均到着間隔 (s)	1.0 1.0 120.0	1.0 30.0*G 250.0
QoS 要求: デッドライン	同期: A (s) 応答: R (s) 目標 QoS: P, Q	240.0 90.0 0.95, 0.8	300.0 60.0 0.95, 0.8
ポーリング 間隔 (s)	通常モード 節約モード	94.7 112.5	63.2 75.0

どうしが交換するメッセージが XML 形式であるため、パーサが必要となる。JAR 容量の制限から、MGO のコードサイズは 2G, 3G で 10 KB, 80 KB が上限となる。MSH と Grid プロキシでは Java 環境が異なるため、MGO について J2ME と J2SE 環境用のコードを用意する。そのため、PH や GH 上で動作するコードは、サイズ制約なしに開発可能である。

##### 4.2 モバイルサーバミドルウェア

MGO の QoS 要求は ADF (Application Descriptor File) に記述され、MSM と MGO が統合されたアプリケーションの起動の際に、QoS ポリシ管理部がその要求を読み込む。表 1 に、実装したブックマーク共有とフォトログアプリケーションの各パラメータを示す。ブックマーク共有は、データ量が小さく、更新頻度が早いアプリケーション、フォトログはデータ量が大きなアプリケーションに対応する。また、ブックマークは集約の効果が大きいアプリケーションとして、複数の更新がつねに 1 KB のメタデータと 1 KB のバイナリデータに集約されて同期されるものとした。一方、フォトログについては、1 KB のメタデータと集約回数  $G$  分のバイナリデータが同期される (表の  $30 * G$ )。同期時間と応答時間のデッドライン要求について、更新の頻度の早いブックマークの同期時間要求を短く設定したのに対し、フォトログは CH に対する応答性を重視して、応答時間要求を短く設定した。通常モードと節約モードのポーリング間隔は、スケジューラのアルゴリズムに基づき、応答デッドライン  $R$  と、目標 QoS の  $P$  と  $Q$  から計算される。

また、MSM の端末リソース監視部は、DoJa API の PhoneSystem クラスを通じて、バッテリー状態および接続状態 (圏外か圏内) を測定する。ただし、API の仕様上、取得できるバッテリー状態は、携帯機のバッテリー残量表示がフル、フルでない、充電中という 3 つの状態に制限されるため、“フルでない” 状態をバッテリー節約モードに対応づけることとした。バッテリー節約のため、MSM は通常モードかバッテリー節約モードか

にかかわらず、アイドル状態（通信や処理を行っていない、次のイベントを待ち受けている状態）の際、sleep API を呼び出して最大 3 秒間スリープするように実装した。スリープの際、アプリケーションのすべてのスレッドの実行は停止され、携帯機は端末固有の省電力状態に移行し、3 秒後に起きる。ただし、同期やポーリングのスケジュールを邪魔しないようにスリープ時間を調節する。

#### 4.3 Grid プロキシと MGM

J2SE をサポートする PC (Intel Pentium4 2.80 GHz, 1 GB メモリ) 上に Grid プロキシと MGM を実装した。Grid プロキシ上には、ポーリングと e メールベースのプッシュをサポートするセッション管理と同期モジュールを実装している。一方、MGM には以下にあげる MGO の管理機能を Globus Toolkit (GTK) を利用して実装した。移送機能は GTK の RFT (GridFTP) を利用して、MGM 間で XML ファイル化された MGO のデータを高信頼にコピーするようにしている。また、MGM の負荷分散機構 (Load Balancer) は GTK の MDS (Monitoring and Discovery Service) から各 Grid ノードの負荷情報を収集して、軽負荷の Grid ノードへ MGO を移送することを可能にしている。さらに、GTK の GRAM (Grid Resource Allocation Manager) により、Grid ノード上に MGM を動的に遠隔実行可能としており、MGO の増加に対するスケーラビリティを向上させることができる。

- 生成・破棄：アプリケーション種別（フォトリログ等）を指定した MGO の生成と、MGO の一意の名前（識別子）の指定による MGO の破棄。
- 名前解決：MGO の一意の名前から、Grid 上の MGO の位置を url 形式で獲得。
- 移送：MGO の実行をサスペンドして MGO のメタデータを XML フォーマットにシリアルライズした後、移送先の Grid ノード上の MGM へメタデータとバイナリデータをコピーし、MGO の実行を再開。

### 5. 性能評価

適応 QoS フレームワークの有効性と携帯機上でのモバイル Grid アプリケーションの実用性を評価するため、プロトタイプの同期時間、応答時間、稼働時間を測定した。パラメータ変更以外の影響を最小化するため、無線リンクの接続性が安定した静止環境で実験を行った。なお、2G の携帯機の上り、下りの最大リンク速度（理論値）は 9.6 Kbps, 28.8 Kbps, 3G ではそれぞれ 64 Kbps, 384 Kbps である。

表 2 転送時間 (msec)

Table 2 Transfer time.

データサイズ (KB)	1	5	10	50
アップロード	2G 2,755	9,140	16,705	85,110
	3G 1,440	2,726	4,882	26,314
ダウンロード	2G 1,365	3,265	6,830	34,870
	3G 1,452	2,532	3,786	13,492

表 3 処理時間 (msec)

Table 3 Processing time.

メタデータサイズ (KB)	1	2	4
パース	2G 185	460	835
	3G 92	236	498
エンコード	2G 5	45	115
	3G 4	12	38

#### 5.1 同期時間

MSM と Grid プロキシ間の同期手順は、エンコード、転送、パースの 3 つの手順から構成される。現在の実装では、差分の抽出や衝突の解決はアプリケーションに任せている。表 2, 表 3 にメタデータのパース、エンコードにかかる処理時間と転送時間を示す。Grid プロキシ上の更新を MSM と同期する際、MSM の同期モジュールは、HTTP 接続を通じてメタデータをダウンロードし、パースして変数や状態を獲得した後、ローカルの MGO に渡す。逆方向の同期の場合、同期モジュールは、MGO の変数や状態をテキスト形式にエンコードして Grid プロキシにアップロードする。バイナリデータは、エンコードやパースなしに送受信する。

表 2, 表 3 の結果を利用することで、スケジュール部は更新データのサイズに基づき、同期時間  $s$  を見積もり、集約時間  $a$  を調整できる。たとえば、2G, 3G で 1 KB のメタデータと 10 KB のバイナリデータを同期する場合、それぞれ  $s = 19.48$ ,  $s = 6.32$  と計算できる。なお、Grid プロキシにおけるパース時間は無視できるほど小さいため、計算には含めていない。

#### 5.2 応答時間

表 1 のポーリング間隔設定で、ブックマーク共有アプリケーションの応答時間を 50 回測定し、応答デッドライン要求を満足するかを検査した。具体的には Grid プロキシが、CH からの要求を受け取ってから MSH のポーリングメッセージを受け取るまでの経過時間が  $R$  に収まる割合を測定した。その結果、2G, 3G の携帯機の両者とも通常時の設定で 0.95, 節約時で 0.81 の QoS 保証率を達成し、目標 QoS 保証率 0.95, 0.8 を満足していることを確認した。

#### 5.3 持続時間

図 6 に、表 1 のポーリング間隔設定におけるアプ

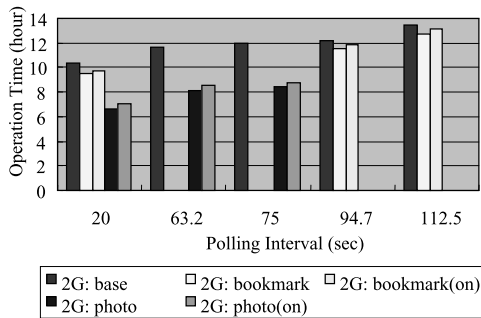


図 6 稼働時間

Fig. 6 Operation time.

リケーションの持続時間を示す。ポーリング間隔は、QoS 要求を考慮しない参照値として 20 秒と、通常モードと節約モードの場合の値（ブックマークの場合は 94.7, 112.5, フォトログの場合は 63.2, 75.0）に設定した。持続時間として、2G の携帯機をフルに充電した後、バッテリー残量表示がフルからフルでない状態に変化するまでの時間を測定した。base は、MSH が待ち受けの場合であり、Grid プロキシへのポーリング以外の通信は行わない。bookmark, photolog はそれぞれのアプリケーションに、表 1 の到着間隔でポアソン到着モデルに従って更新データを生成させ、固定の集約時間（20 秒）で同期を行う場合の持続時間を示す。bookmark(on), photolog(on) は、QoS 要求に基づき集約時間を設定した場合である。

実験の結果、ポーリング間隔の増加によって持続時間が 12% から最大 35% まで拡大された。これにより、アプリケーションの QoS 要求に応じたポーリング間隔設定と、節約モードにおけるポーリング間隔の増加設定が、MSH の稼働時間を延長できることを確認できた。一方、集約時間の調整による持続時間の拡大の効果も 2.7% から最大 5.4% まで確認できた。なお、3G 携帯機でも同様の傾向が観測された。

#### 5.4 リンク品質への適応性

最後に、プロトタイプを測定して得られた処理時間や転送時間等の性能値をシミュレータに入力し、異なる切断パターンに対する同期スケジューリングアルゴリズムの適応性を評価した。離散シミュレータである OMNet++<sup>9)</sup> を利用してモデルを構築し、ポアソン到着モデルに従って平均到着間隔を 180, 300, 600 秒に設定して切断イベントを生成させた。平均切断時間は指数分布に従って 30 秒と 10 秒に設定した。表 1 に示すフォトログアプリケーションの QoS 要求やパラメータを利用して、シミュレーションを行った。

図 7 に、同期デッドライン要求の保証率を示す。

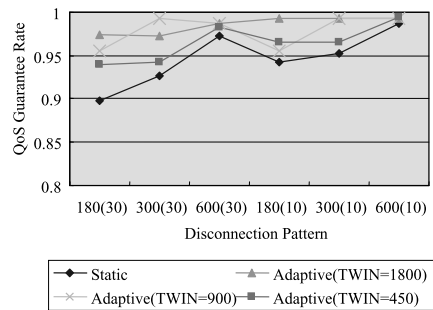


図 7 切断パターンに対する適応性

Fig. 7 Adaptiveness to disconnection patterns.

Static はリンク接続品質を考慮せず、固定的に集約時間を最大 ( $A - s$ ) に設定する方式であり、リンク品質が悪化している環境では、目標 QoS 保証率 0.95 を満足できない。一方、Adaptive はスケジューラが集約時間を動的に設定する方式であり、リンク接続品質を評価するタイムウィンドウ (TWIN) を、3 種類に変化させている。タイムウィンドウの大きさはストレージを圧迫するため、小さくするのが好ましいが、450 秒に設定した場合も QoS を保証できない。一方、タイムウィンドウを 900 秒に設定し、スケジューリングを行うことで同期デッドライン要求の QoS を保証できることが確認された。本評価により、異なるリンク品質環境に対するスケジューラの集約時間の適応アルゴリズムの有効性と、タイムウィンドウ設定が適応アルゴリズムに影響を与えることを確認した。

## 6. ま と め

本論文では、モバイル Grid システムにおける携帯機用のサーバ化ミドルウェア (MSM) に対して適応 QoS フレームワークを導入した。MSM は、同期と応答に関するデッドライン要求、目標 QoS 保証率、バッテリー節約による稼働時間向上等の QoS 要求を、同期処理における集約時間の調節とポーリング間隔の設定の制御に反映する QoS 機構を提供する。可用性・応答性と持続性の間のトレードオフのバランスをとるとともに、バッテリー残量の変化や無線リンクの接続品質の変化に対して適応することが可能である。MSM のプロトタイプを J2ME 携帯機上に実装し、フレームワークが、アプリケーションが指定した目標の QoS 保証率を維持しつつ、稼働時間を拡大できることを確認した。

今後の課題は、モバイル Grid システムの安全性と信頼性の向上である。現状のシステムは携帯機の ID とパスワードにより、Grid プロキシが携帯機を認証する機構を実装済みであるが、ヘテロジニアスなモバイ



ル Grid システムのためのアクセス制御やノード間の信頼関係の管理の機構に取り組んでいく。また、異なるセキュリティポリシーを持つ MGO を Grid ノード上で安全に隔離するためのドメイン分離機構，セキュアな同期および移送プロトコルの構築に取り組んでいく。

### 参 考 文 献

- 1) Foster, I. and Kesselman, C.: The Globus Project: A Status Report, *Proc. Heterogeneous Computing Workshop*, pp.4-18 (1998).
- 2) Clarke, B. and Humphrey, M.: Beyond the Device as Portal: Meeting the Requirements of Wireless and Mobile Devices in the Legion Grid Computing System, *2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing* (Apr. 2002).
- 3) Phan, T., Huang, L. and Dulan, C.: Challenge: Integrating Mobile Wireless Devices Into the Computational Grid, *International Conference on Mobile Computing and Networking (ACM MobiCom2002)*, pp.271-278, ACM Press (Sep. 2002).
- 4) Gonzalez-Castano, F.J., Vales-Alonso, J., Livny, M., Costa-Montenegro, E. and Anido-Rifon., L.: Mobile Interfaces to Computational, Data and Service Grid Systems, *ACM Mobile Computing and Communication Review*, Vol.6, No.2, pp.18-27 (2002).
- 5) Arora, C.W.A. and Pabla, K.S.: JXTA J2ME Implementation Project (2003).  
<http://jxme.jxta.org/>
- 6) Joseph, A.D., Tauber, J.A. and Kaashoek, M.F.: Mobile Computing with the Rover Toolkit. *IEEE Trans. Comput.*, Vol.46, No.3, pp.337-352 (1997).
- 7) Phan, N.A., Chen, I.R. and Yen, I.L.: Algorithms for supporting disconnected write operations for wireless web access in mobile client-server environments, *IEEE Trans. Mobile Computing*, Vol.1, No.1, pp.46-58 (2002).
- 8) Mohapatra, S. and Venkatasubramanian, N.: Parm: Power aware reconfigurable middleware, *IEEE International Conference on Distributed Computer Systems (ICDCS-23)*, (May 2003).
- 9) Varga, A.: The omnet++ discrete event simulation system, *The European Simulation Multiconference (ESM'2001)* (2001).

(平成 17 年 1 月 31 日受付)

(平成 17 年 7 月 4 日採録)



太田 賢 (正会員)

昭和 46 年生。平成 6 年静岡大学工学部情報知識工学科卒業。平成 8 年同大学大学院修士課程修了。平成 10 年同大学院博士課程修了。博士 (工学)。平成 11 年 NTT 移動通信網 (株) 入社。現在 (株) NTT ドコモマルチメディア研究所勤務。平成 9 年度日本学術振興会特別研究会特別研究員。モバイルコンピューティング，端末セキュリティ，マルチメディア通信，分散システムに関する研究に従事。訳書『コンピュータネットワーク第 4 版』(プレントリスホール出版)等。電子情報通信学会，ACM 各会員。



吉川 貴 (正会員)

平成 11 年慶應義塾大学総合政策学部卒業。平成 13 年同大学大学院修士課程修了。同年 (株) NTT ドコモ入社。以来，モバイルコンピューティングに関する研究に従事。現在，同社マルチメディア研究所勤務。



中川 智尋 (正会員)

平成 10 年東京大学工学部電子情報学科卒業。平成 12 年同大学大学院修士課程修了。同年 (株) NTT ドコモ入社。以来，アドホックネットワーク，P2P ネットワーク，端末セキュリティの研究に従事。現在，同社マルチメディア研究所勤務。電子情報通信学会会員。



稲村 浩 (正会員)

昭和 40 年生。平成 2 年慶應義塾大学大学院理工学研究科計測工学専攻修士課程修了。同年日本電信電話 (株) 入社。分散トランザクションシステム，分散ファイルシステムの研究開発に従事。平成 6~7 年カーネギーメロン大学計算機科学科にて訪問研究員。平成 10 年より NTT ドコモ。モバイル環境におけるトランスポートプロトコルに関する研究開発に従事。電子情報通信学会，ACM 各会員。