

Title	ホームネットワークにおける資源を考慮したタスク配置手法に関する研究
Author(s)	清海, 佑太
Citation	
Issue Date	2010-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8933
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

**ホームネットワークにおける資源を考慮した
タスク配置手法に関する研究**

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

清海 佑太

2010 年 3 月

修 士 論 文

ホームネットワークにおける資源を考慮した タスク配置手法に関する研究

指導教官 丹康雄 教授

審査委員主査 敷田幹文 准教授
審査委員 篠田陽一 教授
審査委員 日比野靖 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

0810020 清海 佑太

提出年月: 2010 年 2 月

概 要

現在、家電機器などを接続したホームネットワークを利用して、様々なサービスが実用化されつつある。従来、サービスを実現するため宅内の機器でタスクが実行されることが必須であったが、近年実用化されつつあるサービスでは、サービス提供者、中間サービス事業者、ホームゲートウェイの各要素上でタスクが実行可能である。しかし各要素の持つ資源はそれぞれ異なるため、タスクを実行することにより資源の枯渇が発生する可能性がある。本研究ではシステム上の各要素の持つ資源とタスクの持つ特性を考慮したタスクの配置先決定手法を提案する。ロードバランサで利用されている負荷分散手法を応用したタスクの配置先決定手法を提案し、シミュレーションによって提案手法が有効であるか評価する。また、現在検討されている様々なホームネットワークサービスに適用可能か検証する。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	1
1.3	本論文の構成	2
第2章	SIモデル	3
2.1	SIモデルの背景	3
2.2	SIモデルのアーキテクチャ	4
2.2.1	SIモデル上の各要素の役割	5
2.2.2	SIモデルの各要素のコンポーネント	6
2.3	SIモデル上で想定されるサービスの実現方法	7
2.3.1	宅内監視サービスの例	8
第3章	負荷分散手法	10
3.1	分散システムにおける負荷の問題	10
3.1.1	負荷分散手法	10
3.1.2	タスク配置手法	10
3.2	ロードバランサにおける負荷分散手法	14
第4章	提案手法	17
4.1	資源使用率の定義	17
4.1.1	計算資源の定義	18
4.1.2	通信資源の定義	18
4.2	配置先決定手法	18
4.2.1	適性スコアの定義	18
4.2.2	資源使用率上限値の定義	19
4.2.3	タスクコストの定義	20
4.2.4	特徴係数の定義	20
4.3	サービス上での提案手法の流れ	22
第5章	シミュレーションによる評価実験	23
5.1	実験目的	23

5.2	シミュレーション概要	23
5.2.1	単一タスクの配置先決定実験の概要	24
5.2.2	複数タスクの配置先決定実験の概要	24
5.3	提案手法の実装	24
5.3.1	実装環境	25
5.3.2	負荷情報の取得	25
5.3.3	実行されるタスク	26
5.4	実験結果	27
5.4.1	単一タスクの配置先決定実験結果	28
5.4.2	複数タスクの配置先決定実験結果	34
第 6 章	考察	40
6.1	タスクコストと特徴係数の有効性	40
6.2	適性スコアの有効性	43
6.2.1	適性スコアとタスクの実行時間の関係	44
6.3	一般的なサービスでの適用性	46
6.3.1	一般的なサービスの紹介	46
6.3.2	一般的サービスで実行される共通のタスク	60
6.3.3	一般的なサービスへの適用	61
6.4	考察のまとめ	63
第 7 章	まとめと今後の課題	64

目 次

2.1	宅内機器の接続形態の例	4
2.2	従来のアーキテクチャ	5
2.3	SI モデルのアーキテクチャ	6
2.4	BE および HGW のコンポーネント	7
2.5	SI のコンポーネント	7
2.6	宅内監視サービスの実現例	8
4.1	タスクコストの計算方法	21
4.2	タスクの実行時刻 t_{n-1} と終了時刻 t_n	21
4.3	カテゴリ別に設定された特徴係数の初期値の例	22
5.1	解析タスクを HGW に配置した場合	25
5.2	解析タスクを SI に配置した場合	25
5.3	実験 1 のタスクコストの変化 (HGW 実行時)	29
5.4	実験 1 の適性スコアの変化 (HGW 実行時)	30
5.5	実験 1 の特徴係数の変化 (HGW 実行時)	30
5.6	実験 1 のタスクコストの変化 (SI 実行時)	31
5.7	実験 1 の適性スコアの変化 (SI 実行時)	31
5.8	実験 1 の特徴係数の変化 (SI 実行時)	32
5.9	実験 2 のタスクコストの変化 (HGW 実行時)	35
5.10	実験 2 の適性スコアの変化 (HGW 実行時)	36
5.11	実験 2 の特徴係数の変化 (HGW 実行時)	36
5.12	実験 2 のタスクコストの変化 (SI 実行時)	37
5.13	実験 2 の適性スコアの変化 (SI 実行時)	37
5.14	実験 2 の特徴係数の変化 (SI 実行時)	38
6.1	実験 1 の特徴係数の比較 (r_{cpu}, r_{mem})	42
6.2	実験 2 の特徴係数の比較 (r_{cpu}, r_{mem})	42
6.3	実験 1 の適性スコアの比較 (r_{cpu}, r_{mem})	43
6.4	実験 2 の適性スコアの比較 (r_{cpu}, r_{mem})	44
6.5	ユビキタスな安心・安全サービス	47
6.6	マルチメディアな安心・安全サービス	48

6.7 ユビキタスな教養・娯楽サービス	50
6.8 健康・長寿なヘルスケアサービス	51
6.9 省エネ・エコなホームマネジメントサービス	52
6.10 快適・ふれあいコミュニケーションサービス	54
6.11 高度な介護サービス利用と負担の軽減	56
6.12 家電などの故障対応コールセンター	57
6.13 緊急地震速報との連携による地震初期対応の自動化	59

表 目 次

3.1	<i>Threshold</i> のデフォルト値	15
3.2	機器 A の状態	16
3.3	機器 B の状態	16
5.1	実験で使用した機器	26
5.2	本実験で使用した MIB 情報	26
5.3	$f_{\tau_{anlz},r}^e$ の持つ特徴係数の初期値	27
5.4	$f_{\tau_{comm},r}^e$ の持つ特徴係数の初期値	27
5.5	資源使用率上限値の初期値	27
5.6	実験 1 における解析タスク τ_{anlz} の実行前に計測された資源使用率	28
5.7	実験 1 における解析タスク τ_{anlz} の配置時に計算された適性スコア	28
5.8	実験 1 における解析タスク τ_{anlz} の初回実行時に計測されたタスクコスト	28
5.9	実験 1 におけるタスクコスト算出後に再計算された特徴係数	28
5.10	実験 1 における実行時間の平均値	33
5.11	実験 2 における両タスクの実行前に計測された資源使用率	34
5.12	実験 2 における解析タスク τ_{anlz} の配置時に計算された適性スコア	34
5.13	実験 2 における解析タスク τ_{anlz} の初回実行時に計測されたタスクコスト	34
5.14	実験 2 におけるタスクコスト算出後に再計算された特徴係数	35
5.15	実験 2 における実行時間の平均値	39
6.1	実験 1 で求められたタスクコストの平均値	40
6.2	実験 2 で求められたタスクコストの平均値	41
6.3	実験別に求められた通信タスク τ_{comm} のタスクコスト平均値の比較	45

第1章 はじめに

1.1 研究背景

近年、家電機器や住宅設備などの機器の高機能化が進み、宅内に敷設されたホームネットワークにこれらの機器が接続されるようになりつつある。ホームネットワークはホームゲートウェイによって外部のネットワークと接続されており、外部から提供される様々なサービスが利用可能である。

従来は宅内に設置されたコントローラによって予め決められたタスクを処理していたが、今後はOSGiのような外部から様々なソフトウェアをダウンロードして実行可能な技術を用いて、ホームゲートウェイ自身が様々なタスクをダウンロードして処理することが可能となる[12]。また、従来のサービスは全てサービス提供者によって記述されていたが、ホームネットワークアーキテクチャの多様化や複雑化が進み、全てのサービス利用者が利用可能なサービスを記述することは困難である。このため、サービス利用者の情報を管理しサービス提供者へサービスの記述をサポートする中間サービス事業者が配置されたシステムの利用が実現しつつある。これにより、サービス提供者は各サービス利用者の持つ詳細な条件を考慮せずにサービスの記述が可能となる[9][8]。

1.2 研究目的

ホームネットワークを利用したサービスを実現するために、サービス提供者、中間サービス事業者、ホームゲートウェイという構成要素上で単数あるいは複数のタスクが実行される。サービスを円滑に実現するためには、要素の持つ資源を考慮することが重要となる。なぜなら、要素の持つ資源はそれぞれ異なるため、同じタスクを実行させた場合で資源の不足が発生する可能性があるためである。また、中間サービス事業者やサービス提供者と比べ、ホームゲートウェイは持っている資源が少ないため、資源に関する情報を適切に管理することが重要である。

本研究では、ホームネットワークを利用したサービスが持つ特徴と要素が持つ資源に応じて、タスクの配置先を決定する手法について提案する。要素の持つ資源とは、メモリ資源やCPU資源などのことである。またサービスの持つ特徴とは、サービスを実現するためのタスクによる資源への負荷の度合いを表したものである。この特徴やサービス提供者、中間サービス事業者、ホームゲートウェイの各要素の持つ資源を考慮して最適な要素を選択し、タスク配置を行うことが重要となる。要素の資源を監視し、タスクの特徴を管

理することで、最適な配置先を決定するための手法について提案する。各要素の持つ計算資源や要素間の通信資源が有効に利用されることにより、ホームネットワーク向けサービスが円滑に実現可能となる。提案手法を評価するため、実際のホームネットワークを利用したサービスを模倣したシミュレーションによる評価実験を行う。また、現在考案されている様々なホームネットワークを利用したサービスを調査し、提案手法が一般的なサービスへ適用可能であるか検証する。

1.3 本論文の構成

本論文は以下の構成となっている。

- 第1章
研究の背景や問題点、目的について説明し、本論文の全体としての流れについての説明を行う。
- 第2章
本論文の研究対象となる SI モデルに関して、概要や内部構成についての説明を行う。
- 第3章
関連技術である負荷分散手法やタスク配置先決定手法、ロードバランサについての説明を行う。
- 第4章
本論文で提案するタスクの配置先決定手法についての説明を行う。
- 第5章
提案手法についての有効性を評価するためのシミュレーション実験についての概要と結果についての説明を行う。
- 第6章
提案手法についての有効性に関する評価を行い、今後の展望や課題についての説明を行う。
- 第7章
本論文の総括を行う。

第2章 SIモデル

2.1 SIモデルの背景

近年、家電機器や住宅設備などの機器の高機能化が進み、宅内に敷設されたホームネットワークにこれらの機器が接続されるようになりつつある。ホームネットワークのアーキテクチャはITU-T J.190において勧告されている[4]。ホームネットワークには、固有プロトコルドメインとIPドメインの二つのドメインが導入されている。固有プロトコルドメインは、デジタルAV機器を接続するネットワークや白物家電を接続するネットワークなどの用途別に定められたネットワークに対応するものであり、用途や接続機器の種類によって複数のプレーンに分類される。また、ホームネットワークに接続される機器は、HA (Home Access), HB (Home Bridge), HC (Home Client), HD (Home Device) の4つの機器クラスに分類されている。

Home Access (HA) アクセスネットワークとホームネットワークのIPドメインを接続する機器 (ホームゲートウェイ)

Home Bridge (HB) IPドメイン内のブリッジングやルーティングを行う機器 (ハブ, ルータ)

Home Client (HC) IPドメインと固有プロトコルドメインを接続する機器 (STB, PC, 白物家電のコントローラ)

Home Device (HD) 固有プロトコルドメインの各プレーンに接続される機器 (ECHONET対応の白物家電, IEEE1394を搭載したAV機器)

ホームネットワークはホームゲートウェイによって外部のネットワークと接続されており、外部から提供される様々なサービスが利用可能である。従来は宅内に設置されたコントローラ (HC) によって予め決められたタスク进行处理し、家電機器などの機器 (HD) を制御することでサービスを実現していた。ホームゲートウェイ (HA) はアクセスネットワークとホームネットワークを接続するためだけに利用している機器であった。図 2.1 では、IPドメインの他にデジタルAVプレーン、暮らし環境プレーンの二つのプレーンで構成されているホームネットワークの例を示す。

しかし今後は、ホームゲートウェイ自身がOSGi[5]のように外部から様々なソフトウェアをダウンロードして実行可能な技術を用いて、様々なタスクをダウンロードして処理す

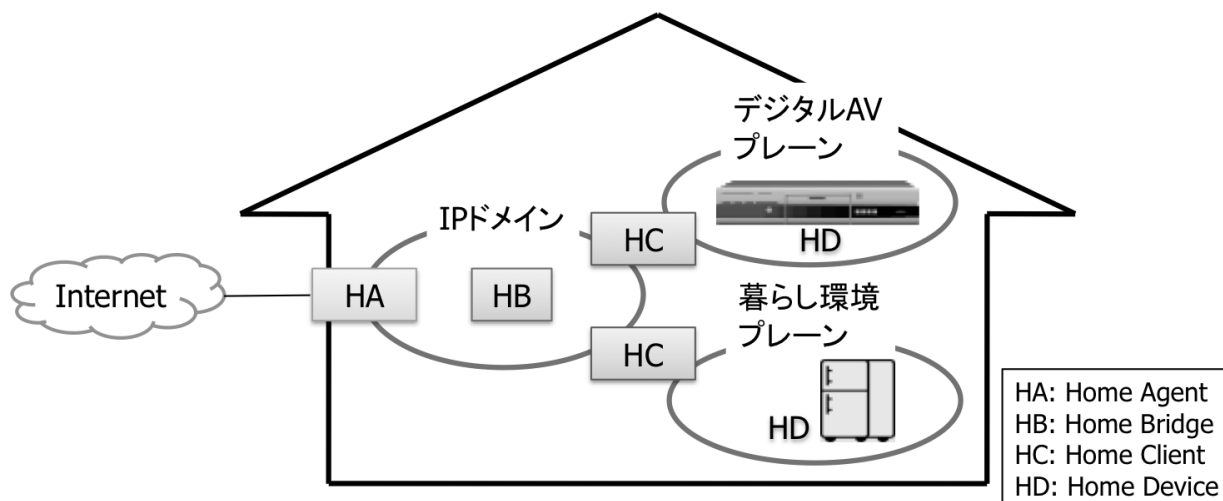


図 2.1: 宅内機器の接続形態の例

ることが可能となる。OSGi はソフトウェアを動的に追加したり実行可能とする Java フレームワークである。

サービスを実現するためには、宅内に設置されている機器や環境などの情報を考慮してサービスを記述する必要がある。このためサービスを記述するサービス提供者は、宅内情報を取得している必要がある。しかし、家庭が持つ家電機器や住宅設備などは各家庭毎に異なり、サービスを利用したい全てのサービス利用者の宅内情報を全て把握することは困難である。また、サービスを利用したい場合はサービス提供者を選択し、サービス提供者が指定した機器を用意し、各自で設定を行う必要がある。これは機器の操作ができるユーザは利用可能であるが、操作が苦手なユーザは自由にサービスを利用することが困難になってしまう (図 2.2)。

そのため、宅内情報やホームゲートウェイに配布するタスクを管理したり、宅内の家電機器や住宅設備などを管理する中間サービス事業者が設置されたシステムが実現されつつある。このシステムを本研究では SI モデルと定義する。現在、誰でも自由にホームネットワーク向けサービスを利用できるように、SI モデルを利用したサービスの実現方法が普及しつつある。

2.2 SI モデルのアーキテクチャ

SI モデルは、バックエンドサービス提供者、中間サービス事業者、ホームゲートウェイ (サービス利用者) の 3 つの要素で構成されている。バックエンドサービス提供者はサービスを記述する役割を持つ要素である。中間サービス事業者は宅内に設置された機器やサービスの管理を行う要素である。また、サービス利用者はホームネットワークに接続された機器を利用してサービスを受ける人のことであり、宅内の機器の管理やサービスの制

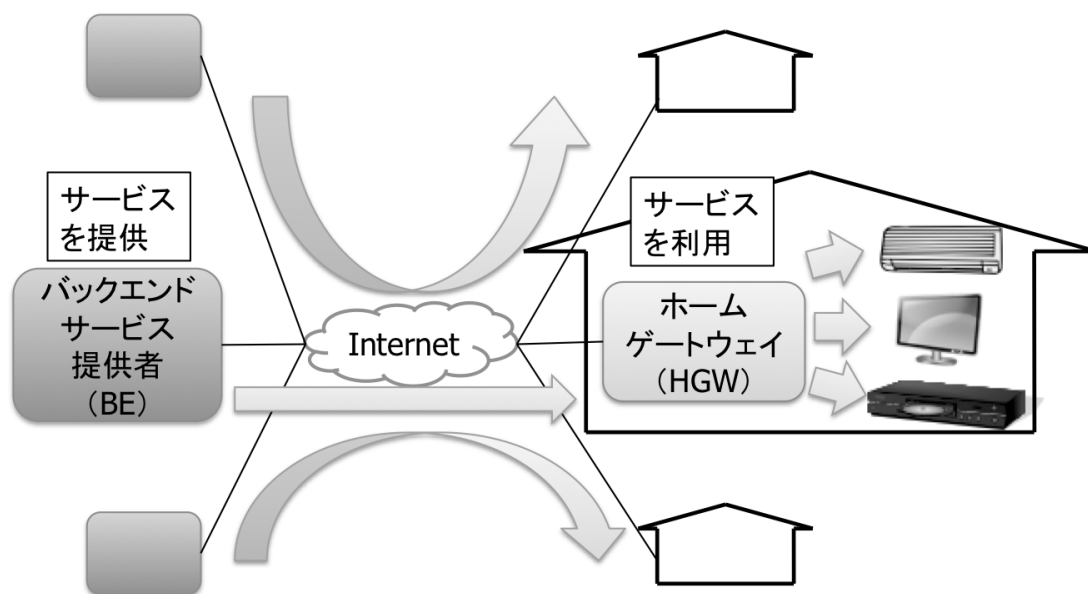


図 2.2: 従来のアーキテクチャ

御についてはホームゲートウェイが行う。SI モデルを利用することにより、サービス提供者は機器のメーカーや宅内の詳細状況を考慮せずにサービスを記述することが可能となり、サービス利用者は機器のメーカーに偏らずにサービスを利用可能となる。SI モデルのアーキテクチャを図 2.3 に示す。

2.2.1 SI モデル上の各要素の役割

バックエンドサービス提供者 (BE: BackEnd service provider)

サービス提供者 (BE) はホームネットワーク向けサービスを提供する役割を持つ要素である。サービスは中間サービス事業者が提供される API を用いて記述される。記述されたサービスは一般的な機器に対する制御が記述され、これが中間サービス事業者によって各サービス利用者が持つ機器に対する制御コマンドに書き換えられる。これにより、サービス提供者は汎用性のある形でサービスを提供することが可能となる。

中間サービス事業者 (SI: Service Intermediary)

中間サービス事業者 (SI) は、BE が記述したサービスを各サービス利用者が持つ具体的な機器に対する制御コマンドに書き換えて、サービスを提供する役割を持つ要素である。SI は各家庭の宅内情報をホームゲートウェイから、サービスに関する情報を BE から取得して管理している。

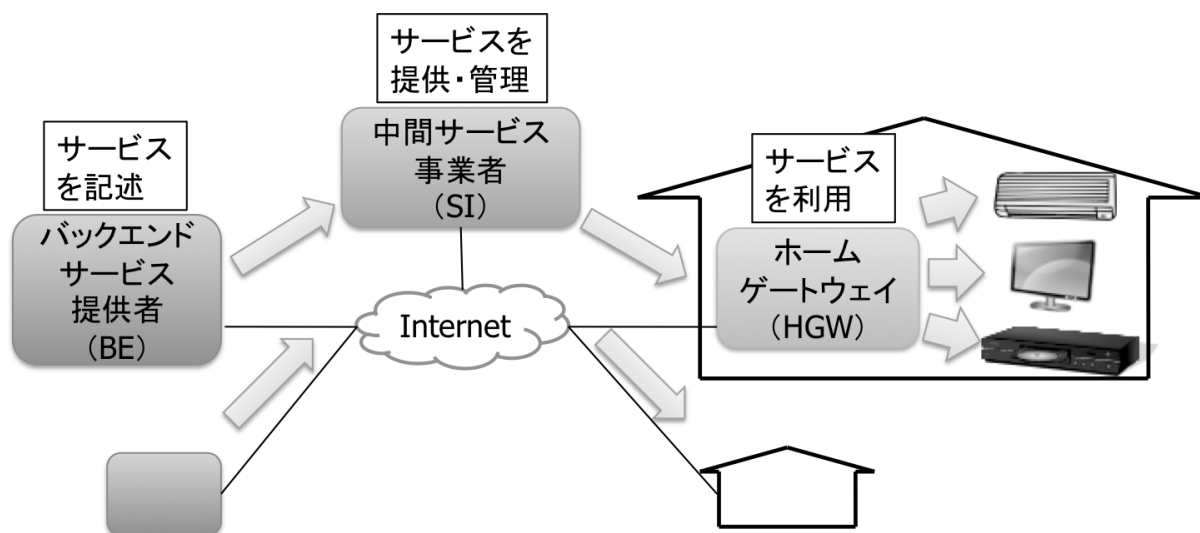


図 2.3: SI モデルのアーキテクチャ

また、サービスを実現するためのタスクを実行する資源を提供しており、各サービス利用者に対して様々なサービスを実現させる役割を持つ。

ホームゲートウェイ (HGW: Home GateWay)

ホームゲートウェイ (HGW) は、サービスを実現するために SI から配布されたタスクを実行する役割を持つ。また、宅内に設置された情報家電や住宅設備などの機器に対して命令を行うほか、それらの機器に関する情報を取得する役割を持つ。

2.2.2 SI モデルの各要素のコンポーネント

次に SI モデル上の各要素のコンポーネントについて説明する。SI と BE, HGW のコンポーネントはそれぞれ異なる。

BE および HGW の持つコンポーネント

まず BE および HGW の持つコンポーネントについて説明する。BE, HGW は同じコンポーネントを持ち、タスク実行プラットフォーム、資源監視エージェントで構成されている。タスク実行プラットフォームはタスクを実行するための基盤である。資源監視エージェントは自要素の持つ資源に関する情報を収集し、SI へ通知する役割を持つ。BE および HGW の持つコンポーネントを図 2.4 に示す。

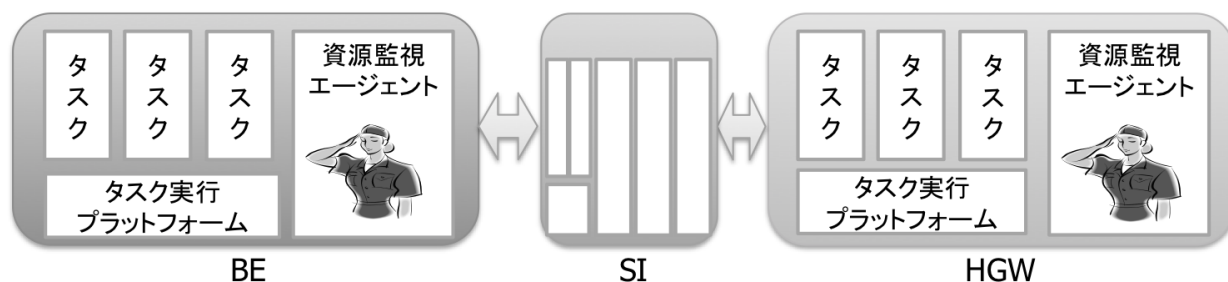


図 2.4: BE および HGW のコンポーネント

SI の持つコンポーネント

次に SI の持つコンポーネントについて説明する。SI の持つコンポーネントはタスク実行プラットフォーム、資源監視エージェントの他に、タスクマネージャ、資源監視マネージャで構成されている。タスクマネージャはタスクの配置先やタスクの持つタスクコストについて管理する。また資源監視マネージャは各要素の持つ資源の状態について管理する。SI の持つコンポーネントを図 2.5 に示す。

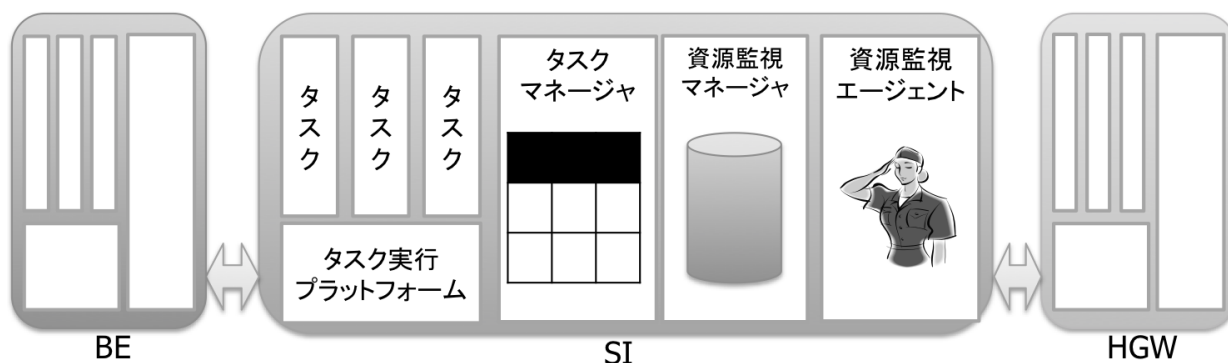


図 2.5: SI のコンポーネント

2.3 SI モデル上で想定されるサービスの実現方法

ホームネットワークを利用したサービスは、宅内に設置されている機器やそれらと接続された機器がタスクが実行されることで実現される。タスクはサービスを実現するために要素上で実行されるソフトウェアである。サービス利用者がサービスを利用したいとき、HGW を経由して、SI へ利用したいサービスを要求する。SI はサービスを実現するためのタスクを選択し、タスクの配置先を決定する。配置先の要素は SI からタスクをダウンロードし、タスクが実行されることでサービスが提供される。

以下にサービスの実現例として宅内監視サービスが実現される場合の各要素の振る舞いについて説明する。

2.3.1 宅内監視サービスの例

宅内監視サービスは、宅内に複数設置されたカメラやセンサから取得したデータを集約し、監視会社へ転送されることで監視会社が宅内の状態を監視するサービスである。家庭内に敷設されたホームネットワークにホームゲートウェイやカメラ、センサなどの全ての機器が接続されている。

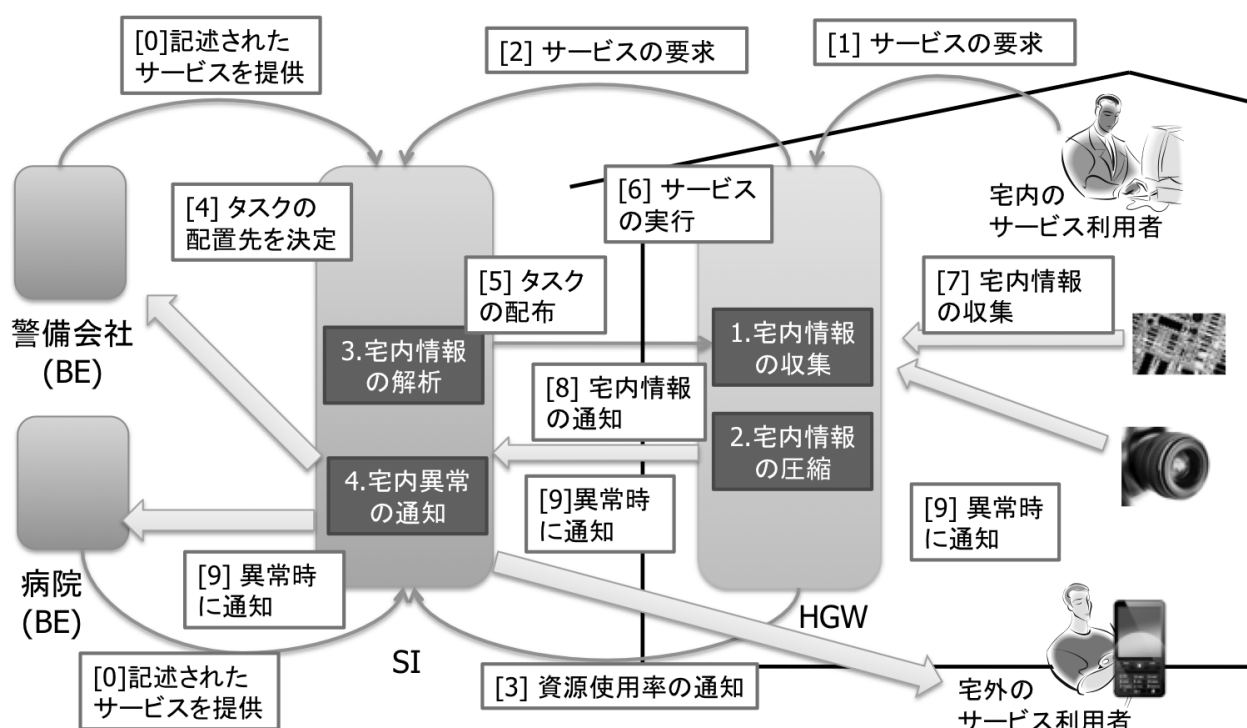


図 2.6: 宅内監視サービスの実現例

タスク₁ : センサやカメラからデータの収集

タスク₂ : 収集した情報の解析

タスク₃ : 宅内情報を外部 (BE (警備会社) や SI など) へ通知

タスク₄ : 宅内情報を圧縮

タスク₁ は、宅内に設置されているセンサでセンシングされた情報を収集したり、カメラで撮影された映像を収集するタスクである。タスク₂ は、収集したセンサ情報やカメラ

映像から不審者を検知や宅内の状態を収集する。タスク₃は、警備会社やストレージサーバのある SI、サービス利用者の持つ携帯電話などの外部へ宅内にある情報を送信する。タスク₄は、収集した宅内情報を圧縮する。

次に宅内監視サービスが実現される流れについて説明する。宅内監視サービスは以下の手順で実現される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（宅内情報の圧縮）を配布し、SI はタスク 3（宅内情報の解析）とタスク 4（宅内異常の通知）を配置
6. タスクが実行されサービスが開始
7. HGW 上のタスク 1 は、宅内の機器（カメラやセンサなど）からデータを収集
8. HGW 上のタスク 2 は収集したデータを圧縮し、HGW から SI へ送信
9. SI 上のタスク 3 は収集したデータを解析し情報を抽出、異常を検知した場合はタスク 4 が警備会社 (BE) や病院 (BE)、宅内機器へ通知

警備会社は SI から提供されている API を利用して宅内監視サービスを記述し、SI へ送信される。サービス利用者は宅内監視サービスを利用したいとき、HGW を経由して SI へサービスの利用を要求する。このとき HGW は自身の持つ資源の情報を SI へ通知する。SI は HGW の持つ資源の情報や SI の持つ資源の情報を考慮して、タスクの配置先を決定する。そして SI からタスクが配布され、それを実行してサービスを実現する。HGW はサービス利用者の宅内に設置されたセンサやカメラから情報を収集し、SI へ宅内情報として送信される。もし宅内の異常を検知したら BE やサービス利用者へ通知される。

本研究では、4 で行われている実行されるタスクの配置先を決定する手法について提案する。

第3章 負荷分散手法

本研究と関連性の高い技術について関連技術について調査を行った。3.1節では負荷分散に関する関連研究の調査、3.2節ではF5 Netoworks社のBIG-IPと呼ばれるロードバランサについて調査を行った。

3.1 分散システムにおける負荷の問題

3.1.1 負荷分散手法

負荷分散とは、分散システム上で負荷がなるべく均等になるように処理を分散して割り当てる手法である[1]。分散システムは、投入された仕事（タスク）をシステム上の複数の計算機（ノード）に分散させて処理させるシステムの枠組みである。分散システム上で負荷分散を実現するために、様々な負荷分散手法について研究されている[6, 7, 10, 11, 16]。

負荷分散方式はシステムの負荷状態に影響されない静的負荷分散方式と、時系列に変化するシステムの負荷状態に適応する動的負荷分散方式がある。さらに動的負荷分散方式には、タスクの実行中にタスクを移動するタスク移送方式と、タスクの実行開始時にタスクの実行先を決定するタスク配置方式がある。本研究ではタスク配置方式に基づいた手法について提案する。

3.1.2 タスク配置手法

本研究で提案するタスクの配置先決定手法は、SIモデル上の要素に対するタスクの配置先決定を行う。これは、計算機の並列分散環境におけるタスクの配置先決定手法（タスクスケジューリング）と関係性が非常に高いと考え、一般的なタスクの配置先決定手法について調査する[2, 13, 15]。

複数のタスクが実行可能であるとき、オペレーティングシステムがどのタスクから実行するかを決定することをタスクスケジューリングといい、ここで使用するアルゴリズムをスケジューリングアルゴリズムという。タスクスケジューリングアルゴリズムとして、以下の手法が存在する。

- Round Robin
- Priority Scheduling

- Multilevel Feedback Queue
- SPTF (Shortest Processing Time First)
- EDF (Earliest Deadline First)
- Bratley
- LDF (Latest Deadline First)
- RM(Rate Monotonic)
- DM(Deadline Monotonic)
- LLF (Least Laxity First)
- ESF (Earliest Start time First)

Round Robin

Round Robin アルゴリズムは、全てのタスクが平等に実行される機会を与えられるアルゴリズムである。Round Robin を実現するために、各タスクにはクオンタムという実行時間の間隔が与えられる。タスクがクオンタムの終了時に実行中の場合、CPU は他の実行可能状態のタスクに与えられる。これにより全てのタスクが公平に実行されるが、タスクの実行切り替えが頻繁に起こるため、それによるオーバーヘッドが大きくなる可能性がある。

Priority Scheduling

Priority Scheduling アルゴリズムは、各タスクに優先順位を割り当て、優先順位の高い実行可能状態のタスクから実行する。重要なタスク（優先度が高いタスク）が早く実行されるという意味では、システムとしての効率性は向上する。しかし、優先度が低いタスクは実行されにくく、優先度の高いタスクが多い場合はいつまでも実行されないことが起こる可能性がある。

Multilevel Feedback Queue

Multilevel Feedback Queue アルゴリズムは CTSS (MIT Compatible Time Sharing System) で使用されていたタスクスケジューリング手法である。Multilevel Feedback Queue は、CPU を消費するタスクに対してクオンタムを長く与え、タスクはクオンタムの長さに応じてクラス分けされる。クラス毎に優先度が与えられ、優先度スケジューリングが行われる。実際には、全てのタスクは実行可能状態になったらクラス 1 に所属する。タス

クが実行され、与えられたクオンタム内でタスクの実行が完了しなかった場合、クラス2に移動させられる。下位のクラスでは優先度は下がるが、与えられるクオンタムは長くなる。実行時間の短いタスクは処理が優先されるが、実行時間が長いタスクは優先度の低い状態で実行されてしまう可能性がある。

SPTF (Shortest Processing Time First)

SPTF (Shortest Processing Time First) アルゴリズムは、処理時間が短く済むタスクを優先的に処理することにより平均ターンアラウンドタイムを短くすることが可能である。しかし完了していないプロセスの終了時間の予測が難しい。解決手法の一つとして経過法という手法があり、過去の実行時間を基に予測値を計算する手法である。これは、現在測定値とこれまでの予測値の加重平均で、次の値を連続して予測する手法である。

EDF (Earliest Deadline First)

EDF (Earliest Deadline First) アルゴリズムは、非周期的に起動されるタスクについてデッドラインの早い順に優先度を付ける手法である。これはリアルタイムスケジューリングの手法の一つであり、リアルタイムシステム上で使用される手法である。リアルタイムシステム上ではタスクのデッドラインが存在しており、EDF ではデッドラインに最も近いタスクから実行される。もし、タスクの実行中に他の優先度の高いタスク（すなわちデッドラインの早いタスク）が実行可能状態になった場合、実行中のタスクを中断して切り替える。このため、EDF は一般的にプリエンプションが可能な条件で使用する手法である。

Bratley

Bratley アルゴリズムは、プリエンプション不可能な条件でも機能するタスクスケジューリング手法である。EDF はプリエンプション不可能な条件では実現することが困難である手法であるのに対し、Bratley アルゴリズムは検索木グラフを利用してタスクの実行に関する全ての順列を生成し、実行可能性を確かめる。実行不可能な順列を除き、実行可能スケジュールが生成されたらそれに従い実行される。

LDF (Latest Deadline First)

LDF (Latest Deadline First) アルゴリズムは、デッドラインから最も遅いタスクから逆順にスケジュールを決める手法である。EDF 手法では、デッドラインが早いタスクから実行していくためデッドラインを守れない場合も存在するが、LDF では最大遅れ時間を最小化するスケジュールを作成することが可能である。

RM(Rate Monotonic)

RM (Rate Monotonic) アルゴリズムは、周期的に実行されるタスクに対して起動周期の短いタスクに優先度を高く設定する手法である。RM アルゴリズムは実行順序などの前後関係の条件がないことや、リソースを取り合うこともなく、あるタスクが実行される次の周期の起動時刻までには前の実行が終了することが条件である。静的スケジューリングにおいて RM アルゴリズムを利用してスケジューリング不可能なタスク周期の場合、他のアルゴリズムを使用しても解決することは不可能であると言える。

DM(Deadline Monotonic)

DM(Deadline Monotonic) アルゴリズムは、RM アルゴリズムの条件と同じときにタスクの実行周期間隔がデッドラインよりも長い場合に拡張したアルゴリズムである。この場合、RM アルゴリズムが起動周期の短いタスクを優先することと比較し、DM アルゴリズムではデッドラインの早いタスクを優先する。

LLF (Least Laxity First)

LLF (Least Laxity First) アルゴリズムは、EDF と同様に動的スケジューリングを EDF と同一の前提条件で行う。EDF と異なるのは、タスクの実行時間とデッドラインの差分から Laxity という時間数を計算し、Laxity の長いタスクから逆順にスケジュールを決めることである。

ESF (Earliest Start time First)

ESF (Earliest Start time First) アルゴリズムは、実行可能なタスクから実行する手法である。時間制約の解決に重点を置くのに対して、このアルゴリズムはリソース制約の解消に重点を置く動的スケジューリング手法である。

タスクスケジューリング手法と提案手法の関連性

これらのタスクスケジューリングは一例であり、他にも様々な環境や条件で利用されるようなタスクスケジューリング手法について研究されている。例えば、ホモジーニアスマルチコア環境でのスケジューリングは、上記のタスクスケジューリングにおいて複数のタスクキューに対する考慮が含まれる。また、ヘテロジーニアスマルチコア環境でのスケジューリングは、さらに処理能力の異なるコアに対する考慮が含まれる。本研究では、能力の異なる複数の処理機能に対してタスクの割り当てを行うことで資源を効率よく使用するという点において、ヘテロジーニアスマルチコア環境でのスケジューリング手法と共通する項目は多い。しかし本研究で提案するタスクの配置先決定手法では、処理機能間に

おけるデータやタスクの転送に要するコストやセキュリティなど、ホームネットワークを利用したサービス特有の問題に対する配慮がなされていることが異なる。

また、デッドラインを保証するかという点も、既存のタスクスケジューリング手法と提案手法とは異なる点がある。リアルタイムシステムでは即時性が非常に重要となるため、デッドラインを守るためのタスクスケジューリング手法が考案される。しかし、ホームネットワークを利用したサービスは即時性も重要であるが、確実に実行されることの方が重要となる。すなわち前者のようなシステムはハードリアルタイムシステムであり、ホームネットワークを利用したサービスはソフトリアルタイムシステムであるといえる。タスクスケジューリング手法では、デッドライン保証の観点から Best effort 型と Guarantee 型、Robust 型と 3 つの種類に分類可能であるが、本研究での提案手法は Best effort 型であるといえる。

3.2 ロードバランサにおける負荷分散手法

ロードバランサは負荷分散を行うための装置であり、主に Web サーバやデータベースサーバなどを利用したシステム上に構築される。主な機能として、サーバに対するクライアントからのコネクション要求を各機器に対して振り分ける。ロードバランサではこの機能を実現するため、様々な負荷分散アルゴリズムを用いて負荷分散を行っている。

F5 Networks 社の BIG-IP シリーズのロードバランサでは、様々な負荷分散アルゴリズムを利用することが可能である [3]、BIG-IP シリーズのロードバランサでは、負荷分散の機能毎に以下のモードが設定可能である。

- Round Robin
- Ratio
- Dynamic ratio
- Fastest
- Least Connections
- Observed
- Predictive

Ratio モードは重み付けラウンドロビンのことであり、重みはユーザにより機器毎に設定される。Dynamic ratio モード Ratio モードも重み付けラウンドロビンであるが、重みは機器の状態を基に一意に設定される。Fastest モードは全てのターゲットの機器に対して ping メッセージを送信し、RTT が最も短い機器を選択する。Least Connections モードは、コネクション数が最小の機器を選択する。Observed モードは、Fastest モードと Least

Connections モードの組み合わせた手法である。つまり、サーバの応答性とコネクション数から相関係数を算出し、その値に応じて機器を選択する。Predictive モードは、Observed mode モードで使用する値に対して傾向からの予測値を算出して機器を選択する。予測値はコネクション数の変化から状態を推測することで得られる値である。

本研究では、BIG-IP で用いられている Dynamic ratio モードを SI モデルに適した形で拡張したタスクの配置先決定手法を提案する。Dynamic ratio では、機器の持つ資源の状態をパラメータを基に機器の重みが計算される。資源として使用するパラメータとして、メモリ使用率、CPU 使用率、ディスク使用率を利用している。これらのパラメータは SNMP (Simple Network Management Protocol) によって取得され、式 (3.1) を利用して資源毎に重み $Weight$ の状態を表す値を計算する。

$$Weight_r = 10^{(Coefficient_r \times \frac{Threshold_r - Utilization_r}{Threshold_r})} \quad (3.1)$$

$Utilization_r$ は、資源 r の資源使用率である。対象となる機器のメモリ使用率、CPU 使用率、ディスク使用率のいずれかが入る。 $Coefficient_r$ は、資源 r に与えられた係数である。デフォルト値では全ての資源に対して 1 という値が設定されている。 $Threshold_r$ は、資源 r に与えられた使用率のしきい値である。しきい値のデフォルト値では以下のように決められている。

表 3.1: $Threshold$ のデフォルト値

資源 r	$Threshold_r$	$Coefficient_r$
メモリ	70	1
CPU	80	1
ディスク	90	1

Dynamic ratio モードでは、各資源毎に式 (3.1) を代入し、資源毎に出力された結果を小数点第一位で四捨五入する。その値の合計値が機器の重みとして決定される。重みの大きな機器が負荷が低いと判断される。

Dynamic ratio モードを利用した計算例として、二つの機器を利用した場合について説明する。機器 A の状態を表 3.2、機器 B の状態は表 3.3 に示す。

この場合、機器 A と機器 B の重みは式 (3.1) を利用することで、以下のように計算される。

表 3.2: 機器 A の状態

資源	$Utilization_r$	$Threshold_r$	$Coefficient_r$
メモリ使用率	35 [%]	70	1
CPU 使用率	18 [%]	80	1
ディスク使用率	45[%]	90	1

表 3.3: 機器 B の状態

資源	$Utilization_r$	$Threshold_r$	$Coefficient_r$
メモリ使用率	10[%]	70	1
CPU 使用率	5 [%]	80	1
ディスク使用率	10[%]	90	1

機器 A の計算結果

$$\begin{aligned}
 10^{1 \times \frac{60}{70}} + 10^{1 \times \frac{75}{80}} + 10^{1 \times \frac{80}{90}} &= 3.16 + 5.96 + 3.16 \\
 3 + 6 + 3 &= 12
 \end{aligned}$$

機器 B の計算結果

$$\begin{aligned}
 10^{1 \times \frac{60}{70}} + 10^{1 \times \frac{75}{80}} + 10^{1 \times \frac{80}{90}} &= 7.20 + 8.66 + 7.74 \\
 7 + 9 + 8 &= 24
 \end{aligned}$$

機器 A は 12, 機器 B は 24 という重みが計算され, 結果より機器 B の方が重みが高く設定された.

第4章 提案手法

本研究では、資源の状態とタスクの特性を基にタスクの配置先を決定する。タスクの配置先はSIモデル上の要素である。本研究ではSI, HGWの二つを利用するが、計算資源として利用可能な計算機がある場合は追加することが可能である。配置先をSIモデル上の要素 e として表す。

$$E = \{e : hgw \cup si\} \quad (4.1)$$

また、実行するタスクは τ として表す。

$$T = \{\phi \cup \tau_i : 1 \leq i \leq N\} \quad (4.2)$$

提案する配置先決定手法は、 E に含まれている要素の資源を基に適性スコアを利用している。次に本研究で扱う資源とタスクに関する情報を定義し、配置先の決定手法について説明する。

4.1 資源使用率の定義

負荷の種類を計算資源にかかる負荷、通信資源にかかる負荷に分別する。本研究では、計算資源としてCPU (r_{cpu})、メモリ (r_{mem})、ディスク (容量 (r_{dsk})、帯域 (r_{dskio})) を利用する。また、通信資源としてネットワークインタフェース (r_{iface}) を利用する。これらの資源を選択した理由は後述 (4.1.1節, 4.1.2節) する。

$$R = \{r : r_{cpu} \cup r_{mem} \cup r_{dsk} \cup r_{dskio} \cup r_{iface}\} \quad (4.3)$$

本研究では、計算資源にかかる負荷を表す指標としてCPU使用率 ($u_{r_{cpu}}$)、メモリ使用率 ($u_{r_{mem}}$)、ディスク容量使用率 ($u_{r_{dsk}}$)、ディスクI/O使用率 ($u_{r_{dskio}}$) を利用する。また、

通信資源にかかる負荷を表す指標としてネットワークインタフェース使用率 ($u_{r_{iface}}$) を利用する.

$$U_R^E(t) = \{u_r^e(t) : e \in E, r \in R, t \in \mathbb{Q}\} \quad (4.4)$$

u_r の計測は一定の間隔で行われるものとし, 本研究では 1 分毎に u_r の計測を行っている.

4.1.1 計算資源の定義

本研究では, 計算資源として CPU, メモリ, ディスクの 3 つを取り扱う. CPU の負荷を表すものとして, load average と CPU 使用率, メモリの負荷はメモリ使用率, ディスクの負荷はディスク容量使用率とディスク I/O 使用率があげられる.

CPU の負荷を示す指標では, load average と CPU 使用率があげられる. load average は CPU の処理を待っているプロセス数を表し, CPU 使用率は CPU 時間のなかで実行中のプロセスが占有している割合を表す. 負荷の低いプロセスが大量に起動している状態では, CPU 使用率は低い値となるが load average では高い値となりうるため, 有効とは言えない. また, load average では実行するプロセス数が増加すると値が無限に増加してしまう. そこで本研究では, 負荷のある範囲の値で表現できる CPU 使用率を利用する.

4.1.2 通信資源の定義

通信資源の状態を取得するための有効な手段としてネットワークインターフェイスの使用率を用いる. これは宅内から通信資源に関する情報を計測するパラメータとして適切であると考えたためである.

4.2 配置先決定手法

4.2.1 適性スコアの定義

適性スコアはタスクの配置先として適性があるか示す指標として利用する. $s_{\tau,r}^e(t)$ は, タスク τ の配置先を決定する時刻 t における要素 e 上での資源 r の適性スコアを示す. 適性スコアは資源使用率やタスクが持つ情報を基に求められ, SI モデル上の機器においてタスクを実行することが適しているかを定量的に示すことができる.

$$S_{T,R}^E(t) = \{s_{\tau_i,r}^e(t) : e \in E, \tau_i \in T, r \in R, t \in \mathbb{Q}\} \quad (4.5)$$

適性スコア $s_{\tau,r}^e$ は以下の式を利用して計算される.

$$s_{\tau,r}^e(t) = 10 \left\lfloor \frac{l_r^e - \{c_{\tau_i,r}^e(t-1) + u_r^e(t)\}}{l_r^e} \right\rfloor \quad (4.6)$$

$u_r^e(t)$ はある時刻 t における要素 e が持つ資源 r の資源使用率を表している. 本研究では 4.1 節で定義した 5 つの資源使用率を利用する.

l_r^e は要素 e が持つ資源の資源使用率の上限値を表しており, 要素が消費してもよい資源使用率の上限値を要素毎に示している値である.

$f_{\tau,r}^e$ はタスク τ の配置先決定時 t における特徴係数であり, タスクがどの資源に対してどれだけ消費するかというタスクの特徴を示す値である.

$c_{\tau,r}^e$ は, タスク τ の過去のタスク実行時 $t-1$ における要素 e が持つ資源 r のタスクコストであり, 資源使用率の変化量 (パーセントポイント) を示す値である.

適性スコアを利用してタスクの配置先を決定する手法について説明する. 各要素の適性スコアを計算することで, 資源毎に適性スコアが算出される. この値をそれぞれ比較し, 適性スコアの大きい方の資源の数が多い要素が総合的に見て資源に余裕があると判断することが可能となり, タスクの配置先として決定される.

4.2.2 資源使用率上限値の定義

l_i^e は, 要素 e の使用可能な資源使用率の上限を表している. 資源毎に資源使用率の上限は異なっているため, 全ての資源に対して同じ上限値を設けてしまうのは有効とは言えない. また, 要素毎に使用可能な資源使用率の上限は異なることが言える. 例えば, SI は複数の HGW へ資源を提供しているため, 一つのサービス利用者が SI の資源を 100% 利用可能ではないと考える. そのため, 各要素で資源毎に上限値を設定するために l_r^e を設定する.

$$L_R^E = \{l_r^e : e \in E, r \in R\} \quad (4.7)$$

4.2.3 タスクコストの定義

タスクコストは、タスクを未実行のときと実行中のときの資源使用率の変化量をパーセントポイントで表したものである。 $c_{\tau,r}^e(t)$ とは、タスク τ の配置先決定時 t における要素 e へ配置する際の資源 r のタスクコストを表す。

$$C_{T,R}^E = \{c_{\tau_i,r}^e : e \in E, \tau_i \in T, r \in R\} \quad (4.8)$$

タスクコストはタスクを実行する毎に値は更新されることで精度を増している。次にタスクコストの求め方について説明する。

タスクコストの導出法

タスクコストの求め方について説明する。タスクコストを求めたいタスクを τ_1 とし、全部で x 個のタスクが実行されているとする。全体の u_r から τ_1 が使用している割合を、後述する特徴係数を基にして求める。実行中のタスクの持つ特徴係数の合計値 $F_{\tau,r,\text{old}}^e$ を以下のように定義し、 $F_{\tau,r,\text{old}}^e$ を利用して $c_{\tau_1,r,\text{new}}^e$ は式 (4.9) のように計算される (図 4.1)。

$$\begin{aligned} F_{\tau,r,\text{old}}^e &= \sum_{i=1}^x f_{\tau_i,r,\text{old}}^e \\ c_{\tau_1,r,\text{new}}^e &= \frac{1}{2} \left[\frac{f_{\tau_1,r,\text{old}}^e}{F_{\tau,r,\text{old}}^e} \times \{u_r^e(t_n) - u_r^e(t_{n-1})\} + c_{\tau_1,r,\text{old}}^e \right] \end{aligned} \quad (4.9)$$

タスクは時刻 t_{n-1} のときに投入され、時刻 t_n のときに終了している (図 4.2)。

直前に実行されたタスクのタスクコストを計測するために、過去の実行時に計測されたタスクコストを利用している。これは、特徴係数の割合だけでは正確なタスクコストを求めることが困難な場合もあるため、過去に計測された情報を基に再計算することにより、計測されたタスクコストの信頼性を上げることが可能なためである。

4.2.4 特徴係数の定義

特徴係数 f はタスクの持つ特徴や条件に応じて値が決定されるパラメータである。 $f_{\tau,r}^e(t)$ とは、タスク τ の配置先決定時 t における要素 e へ配置する際の資源 r の特徴係数を表す。

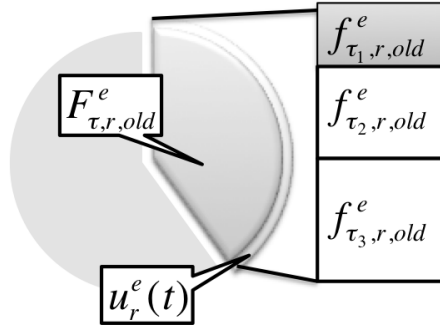


図 4.1: タスクコストの計算方法

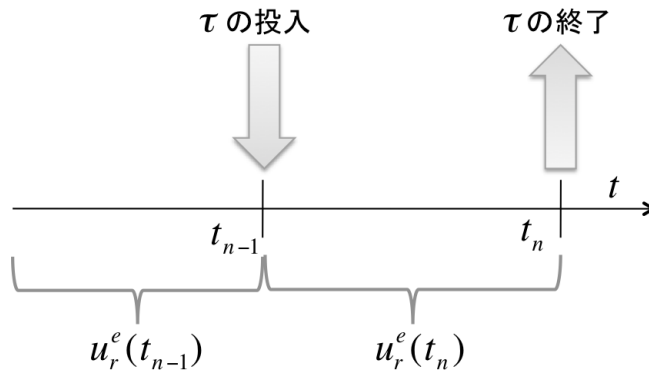


図 4.2: タスクの実行時刻 t_{n-1} と終了時刻 t_n

$$F_{T,R}^E = \{f_{\tau_i,r}^e : e \in E, \tau_i \in T, r \in R\} \quad (4.10)$$

特徴係数の導出法

特徴係数は予め初期値が与えられている。初期値はタスクのカテゴリにより一意に決定されるものとする。カテゴリ毎の特徴係数の例を図 4.3 に示す。

また、特徴係数はタスクコストが計測されたあとに、タスクコストを利用して再計算される。特徴係数の再計算を式 (4.11) に示す。

$$f_{\tau_1,r,\text{new}}^e = \frac{1}{2} \left\{ \frac{1}{100} c_{\tau_1,r}^e(t) + f_{\tau_1,r,\text{old}}^e \right\} \quad (4.11)$$

サービス種類	タスクカテゴリリスト	f_{cpu}	f_{mem}	f_{dsk}	f_{dskio}	f_{iface}
情報収集	センサからデータを取得	0.2	0.2	0.2	0.4	0
	カメラから映像を取得	0.3	0.3	0.3	0.5	0
	機器から状態を取得	0.2	0.2	0.2	0.3	0
情報処理	カメラ映像の圧縮、変換	0.9	0.9	0.4	0.7	0
	センサから情報の抽出	0.9	0.9	0.2	0.7	0
情報通信	コンテンツのダウンロード	0.4	0.4	0.6	0.6	0.6
	外部と映像を送受信	0.8	0.8	0.8	0.8	1
機器制御	宅内の警報機器を制御	0.1	0.1	0.1	0.1	0.1

図 4.3: カテゴリ別に設定された特徴係数の初期値の例

式 (4.11) では、タスクコストの割合 ($\frac{1}{100}c_{\tau_1,r}^e(t)$) と過去の特徴係数 ($f_{\tau_1,r,\text{old}}^e$) との平均値を、新しい特徴係数 ($f_{\tau_1,r,\text{new}}^e$) として計算している。これは、過去の実行結果を反映させることで特徴係数の精度を増し次にタスクコストを計測するときに、より正確なタスクコストを計測するためである。

4.3 サービス上での提案手法の流れ

提案手法を利用したサービスの実現方法を説明する。提案手法を取り入れてサービスを実現するために、以下の手順で作業が行われる。

1. HGW と SI の資源使用率、資源使用率上限値、タスクコストを利用して適性スコアを計算
2. HGW と SI での適性スコアを資源毎に比較
3. 適性スコアの高い資源を多く持っている要素が配置先として決定
4. タスクの配置, 実行
5. タスクの実行終了後にタスクコストが計算
6. タスクコストを基に特徴係数が更新

具体的には、配置したいタスク τ_1 が存在する時、HGW から収集した U_r^{hgw} と SI の持つ U_r^{si} 、また C_r^{hgw} と C_r^{si} を基に、 $S_{\tau_x,r}^{hgw}$ と $S_{\tau_x,r}^{si}$ を算出する。そして、 $S_{\tau_x,r}^{hgw}$ と $S_{\tau_x,r}^{si}$ の値を資源毎に比較し、大きい方の資源の適性スコアを多く持っている要素をタスクの配置先として決定する。

第5章 シミュレーションによる評価実験

本研究で提案するタスクの配置先決定手法の有効性を評価するために、PC を利用したシミュレーションによる評価実験を行った。

5.1 実験目的

シミュレーションでは、本研究で提案するタスクの配置先決定手法が有効なものか評価する。配置するタスクの数や配置するときの要素の状態を変更し、適性スコアがどのように変化するか調査した。

5.2 シミュレーション概要

本研究で提案する手法の評価を行うため、実際のホームネットワークを利用したサービスを想定したシミュレーションによる評価実験を行った。ホームネットワークを利用したサービスを想定したシミュレーションとして、2.3.1 節で解説した宅内監視サービスを模倣した。

宅内監視サービスはセンサから情報を収集する収集タスク、宅内の状態を解析する解析タスク、宅内情報を警備会社などの外部へ通知する通信タスクなどのタスクが実行されることで実現される。

また、適性スコアやタスクコスト、特徴係数の特性を調査するため、各要素上では以下の手順を複数回実行し各パラメータの推移を計測した。また、同時にタスクの実行に必要な時間も計測して適性スコアとの関係も検証した。

1. SI と HGW の資源使用率を取得
2. 適性スコアを算出
3. タスクを実行（実行時間を計測）
4. タスクコストを計測し、特徴係数を再計算

本研究で提案する配置先決定手法が有効であるか検討するため、実行されるタスクの数に応じて二種類の条件で実験を行った。

- 単一タスクの配置先決定実験
- 複数タスクの配置先決定実験

5.2.1 単一タスクの配置先決定実験の概要

単一タスクの配置先決定実験では、一つのタスク（解析タスク τ_{anlz} ）を実行する場合の資源使用率や適性スコア、タスクコスト、特徴係数を計測して配置先を決定する実験を行った。

5.2.2 複数タスクの配置先決定実験の概要

複数タスクの配置先決定実験では、サービスを想定して複数のタスク（解析タスク τ_{anlz} と通信タスク τ_{comm} ）を実行する場合の資源使用率や適性スコア、タスクコスト、特徴係数を計測して配置先を決定する実験を行った。複数のタスクを実行させるため、解析タスクの配置先と同じ要素に対して通信タスクも配置される。ただし、解析タスクの配置先が HGW か SI によって、振る舞いが異なる。各要素での振る舞いを以下に示す。

- HGW で情報解析タスクを実行する場合
 1. 解析タスク τ_{anlz} が実行され、解析前ファイルを読み込んで解析後ファイルを出力
 2. 通信タスク τ_{comm} が実行され、解析後ファイルを送信
- SI で情報解析タスクを実行する場合
 1. 通信タスク τ_{comm} が実行され、解析前ファイルを受信
 2. 解析タスク τ_{anlz} が実行され、解析前ファイルを読み込んで解析後ファイルを出力

通信タスクはどちらの要素でも実行されるが、解析タスクが配置される要素によって転送されるデータの大きさが異なっている。なぜなら HGW で情報解析タスクを実行した場合、通信すべきデータは解析完了後のデータのため容量は小さくなるためである。また SI で情報解析タスクを実行した場合、通信すべきデータは解析完了前のデータのため容量は大きくなる。

5.3 提案手法の実装

本研究で提案するタスクの配置先決定手法の実装方法について説明する。

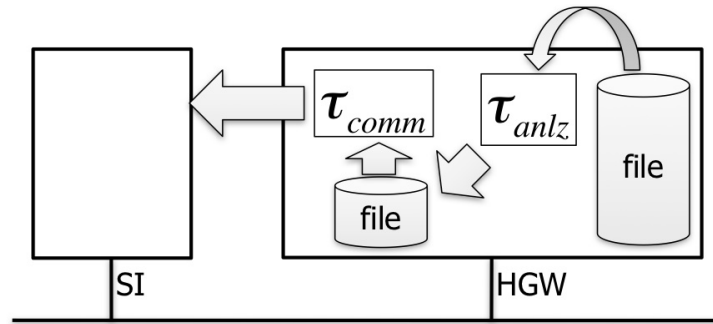


図 5.1: 解析タスクを HGW に配置した場合

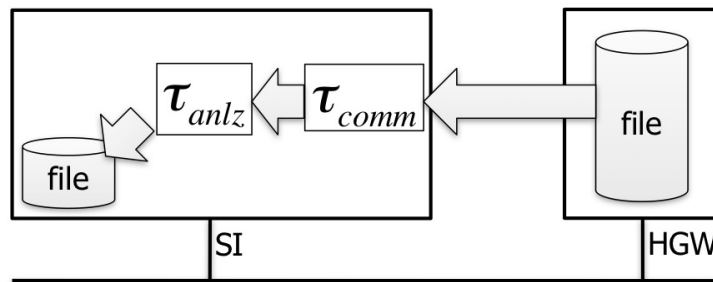


図 5.2: 解析タスクを SI に配置した場合

5.3.1 実装環境

SI と HGW として、2 台のラップトップ PC を利用して実験を行う。本研究では機器の性能が重要となるので、実験に使用する機器の持つ資源の大きさが異なる機器を複数用意した。一般的なホームゲートウェイは PC よりも性能が十分に劣るので、設定により CPU のクロック周波数とメモリ容量の制限を行った。CPU のクロック周波数の制限は `cpufreq` コマンドを利用した。また、メモリ容量の制限は `grub` の設定で行った。

使用する機器の一覧を図 5.1 に示す。

5.3.2 負荷情報の取得

資源使用率を取得するため、SNMP (Simple Network Management Protocol) を利用した。SNMP はネットワークで接続された計算機間でやり取りすることでの情報を取得するためのプロトコルである。SNMP で取得される情報は管理情報ベース (MIB: Management Infomation Base) である。本実験で利用する MIB 情報を表 5.2 に示す。

表 5.1: 実験で使用した機器

資源情報	SI	HGW
CPU	1.70 GHz	600 MHz
メモリ	1.6 GB	256 MB
HDD	60GB	8GB
ネットワーク規格	1000BASE-T	100BASE-TX
OS	Ubuntu 9.10	CentOS 5.3

表 5.2: 本実験で使用した MIB 情報

資源	MIB 情報
r_{cpu}	UCD-SNMP-MIB::ssCpuRawUser
	UCD-SNMP-MIB::ssCpuRawSystem
r_{mem}	UCD-SNMP-MIB::memTotalReal
	UCD-SNMP-MIB::memAvailReal
r_{dsk}	UCD-SNMP-MIB::dskTotal
	UCD-SNMP-MIB::dskUsed
r_{dskio}	UCD-SNMP-MIB::diskIONRead
	UCD-SNMP-MIB::diskIONWritten
r_{iface}	IF-MIB::ifSpeed
	IF-MIB::ifInOctets
	IF-MIB::ifOutOctets

5.3.3 実行されるタスク

実行されるタスクは、SI モデル上で実行されるようなタスクの性質を持たせる必要がある。しかし、SI モデルは一般には普及しておらず、実際に稼働しているホームネットワークを利用したサービスの諸特性は現時点で必ずしも明確とはなっていない。そこで本研究ではタスクの種類に応じて擬似的なソフトウェアを実装し、それをサービスを実現するタスクとみなして PC 上で実行することで評価する。実行するタスクとして、解析タスク τ_{anlz} と通信タスク τ_{comm} を実装した。解析タスク τ_{anlz} として、ファイルを読み込み一文字ずつ走査するプログラムを構築した。また通信タスク τ_{comm} は、ファイルを保持している HGW から SI へファイルを送信する機能とファイルを受信する機能を構築した。送信されるファイルは 5.2.2 節で説明した通り、二種類のファイル (1GB と 100MB) を用意した。

各タスクの持つ特徴係数の初期値を示す。まず、解析タスク $f_{\tau_{anlz},r}^e$ の持つ特徴係数の

初期値を表 5.3 に示す.

表 5.3: $f_{\tau_{anlz},r}^e$ の持つ特徴係数の初期値

	$f_{\tau_{anlz},r_{cpu}}$	$f_{\tau_{anlz},r_{mem}}$	$f_{\tau_{anlz},r_{dsk}}$	$f_{\tau_{anlz},r_{dskio}}$	$f_{\tau_{anlz},r_{iface}}$
HGW	0.4	0.4	0.1	0.3	0.1
SI	0.3	0.3	0.1	0.3	0.1

次に, 転送タスク $f_{\tau_{comm},r}^e$ の持つ特徴係数の初期値を表 5.4 に示す.

表 5.4: $f_{\tau_{comm},r}^e$ の持つ特徴係数の初期値

	$f_{\tau_{comm},r_{cpu}}$	$f_{\tau_{comm},r_{mem}}$	$f_{\tau_{comm},r_{dsk}}$	$f_{\tau_{comm},r_{dskio}}$	$f_{\tau_{comm},r_{iface}}$
HGW	0.2	0.2	0.1	0.3	0.4
SI	0.1	0.1	0.1	0.3	0.2

SI の方が HGW よりも低いのは, SI が持つ資源は HGW の持つ資源よりも十分に大きいため, 負荷の程度は小さくなると考えたためである.

5.4 実験結果

まず, 実験で使用した資源使用率上限値 l_r^e を以下に示す.

表 5.5: 資源使用率上限値の初期値

	$l_{r_{cpu}}^e$	$l_{r_{mem}}^e$	$l_{r_{dsk}}^e$	$l_{r_{dskio}}^e$	$l_{r_{iface}}^e$
HGW	100	100	100	100	100
SI	70	80	90	80	70

資源使用率上限値 l_r^e は, 3.2 節で説明した BIG-IP の Dynamic ratio モードで使用されている初期値を利用した. l_r^{hgw} より l_r^{si} の方が小さい値なのは, HGW は私有される機器なので独占されても問題がないが, SI はサービス利用者に共有される要素なので常にマージンを残す必要があるためである.

次に, 各実験毎に求めた適性スコアやタスクコストなどの結果をまとめる.

5.4.1 単一タスクの配置先決定実験結果

実験1における実験結果として、初回実行時に計測された資源使用率，計算された適性スコア，タスク実行後に計算されたタスクコスト，更新された特徴係数を示す．

まず，解析タスク τ_{anlz} の実行前の HGW 及び SI の資源使用率を表 5.6 に示す，

表 5.6: 実験1における解析タスク τ_{anlz} の実行前に計測された資源使用率

	$u_{r_{cpu}}^e$	$u_{r_{mem}}^e$	$u_{r_{dsk}}^e$	$u_{r_{dskio}}^e$	$u_{r_{iface}}^e$
HGW	1.78	13.0	31.9	3.28	0.00065
SI	0.35	3.39	7.09	3.03	0.95

次に，取得した資源使用率（表 5.6）の値を基に計算される，解析タスク τ_{anlz} を配置する際に計算した適性スコアを表 5.7 に示す．

表 5.7: 実験1における解析タスク τ_{anlz} の配置時に計算された適性スコア

	$s_{\tau_{anlz}, r_{cpu}}^e$	$s_{\tau_{anlz}, r_{mem}}^e$	$s_{\tau_{anlz}, r_{dsk}}^e$	$s_{\tau_{anlz}, r_{dskio}}^e$	$s_{\tau_{anlz}, r_{iface}}^e$
HGW	9.60	7.41	4.79	9.27	10.00
SI	9.89	9.07	8.34	9.25	9.76

次に，解析タスク τ_{anlz} の実行後に計測したタスクコストを表 5.8 に示す．

表 5.8: 実験1における解析タスク τ_{anlz} の初回実行時に計測されたタスクコスト

	$c_{\tau_{anlz}, r_{cpu}}^e$	$c_{\tau_{anlz}, r_{mem}}^e$	$c_{\tau_{anlz}, r_{dsk}}^e$	$c_{\tau_{anlz}, r_{dskio}}^e$	$c_{\tau_{anlz}, r_{iface}}^e$
HGW	38.8	18.6	0.0358	24.2	0
SI	15.7	12.6	0.15	23.1	0

次に，表 5.8 に示したタスクコストを基に再計算された特徴係数を表 5.9 に示す．

表 5.9: 実験1におけるタスクコスト算出後に再計算された特徴係数

	$f_{\tau_{anlz}, r_{cpu}}^e$	$f_{\tau_{anlz}, r_{mem}}^e$	$f_{\tau_{anlz}, r_{dsk}}^e$	$f_{\tau_{anlz}, r_{dskio}}^e$	$f_{\tau_{anlz}, r_{iface}}^e$
HGW	0.394	0.293	0.0158	0.221	0
SI	0.228	0.212	0.0508	0.216	0

タスク τ の配置先を決定する際、資源毎に予め決められた資源使用率上限値 l_r^e と定期的 to 取得された資源使用率 r_r^e (表 5.6 参照) とタスク τ の持つタスクコスト $c_{\tau,r}^e$ (初回実行時なので計算時には全て 0 が代入される) を基にして適性スコア $s_{\tau,r}^e$ (表 5.7) が計算される。タスクの実行後にはタスクコスト $c_{\tau,r}^e$ (表 5.8) が求められ、タスクコストから特徴係数 $s_{\tau,r}^e$ (表 5.9) が再計算された。

以下に、HGW でタスク τ_{anlz} を繰り返し実行した場合に計測したタスクコスト、適性スコア、特徴係数の変化を、図 5.3 から図 5.5 に示す。

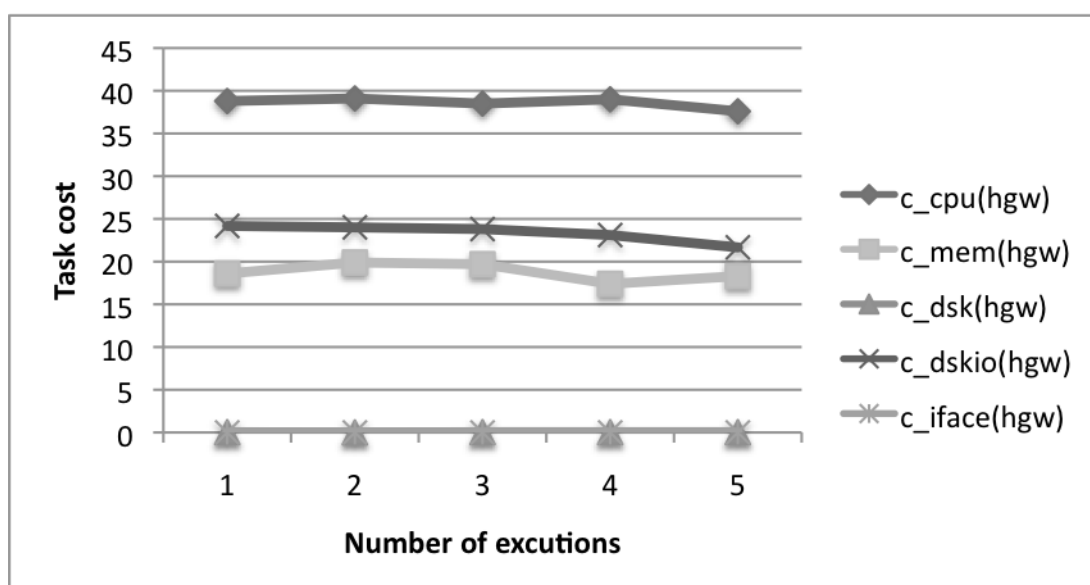


図 5.3: 実験 1 のタスクコストの変化 (HGW 実行時)

図 5.3 は、HGW 上でタスク τ_{anlz} を実行する毎に計測したタスクコストの変化の様子を示している。図 5.5 は、HGW 上でタスク τ_{anlz} を実行する前に計算された適性スコアの変化の様子を示している。図 5.5 は、HGW 上でタスク τ_{anlz} を実行する毎に計算されたタスクコストを基に計算された特徴係数の変化の様子を示している。

次に、SI でタスク τ_{anlz} を繰り返し実行した場合に計測したタスクコスト、適性スコア、特徴係数の変化を、図 5.6 から図 5.8 に示す。

図 5.6 は、SI 上でタスク τ_{anlz} を実行する毎に計測したタスクコストの変化の様子を示している。図 5.8 は、SI 上でタスク τ_{anlz} を実行する前に計算された適性スコアの変化の様子を示している。図 5.8 は、SI 上でタスク τ_{anlz} を実行する毎に計算されたタスクコストを基に計算された特徴係数の変化の様子を示している。

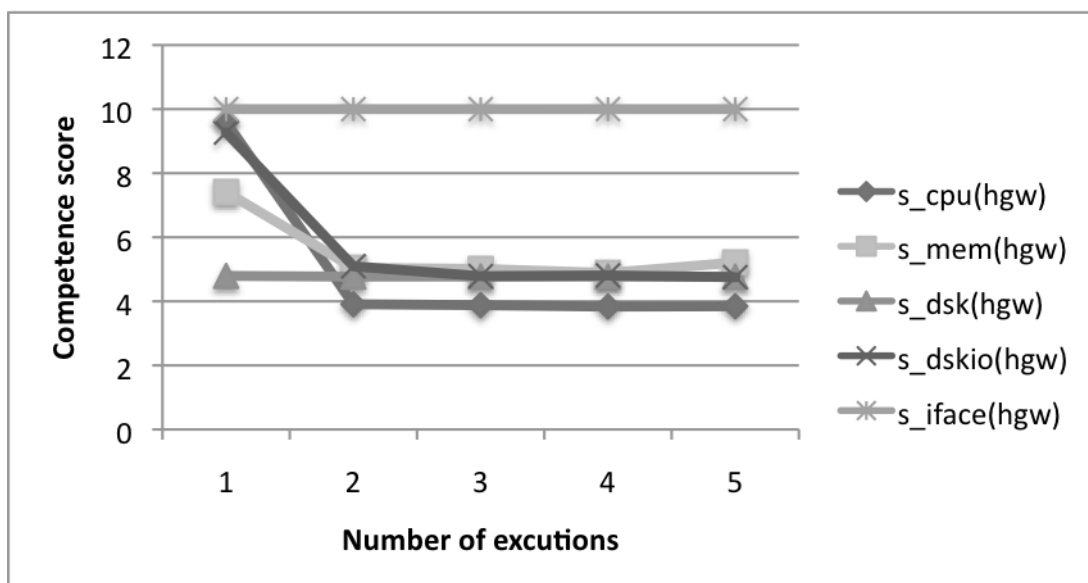


図 5.4: 実験 1 の適性スコアの変化 (HGW 実行時)

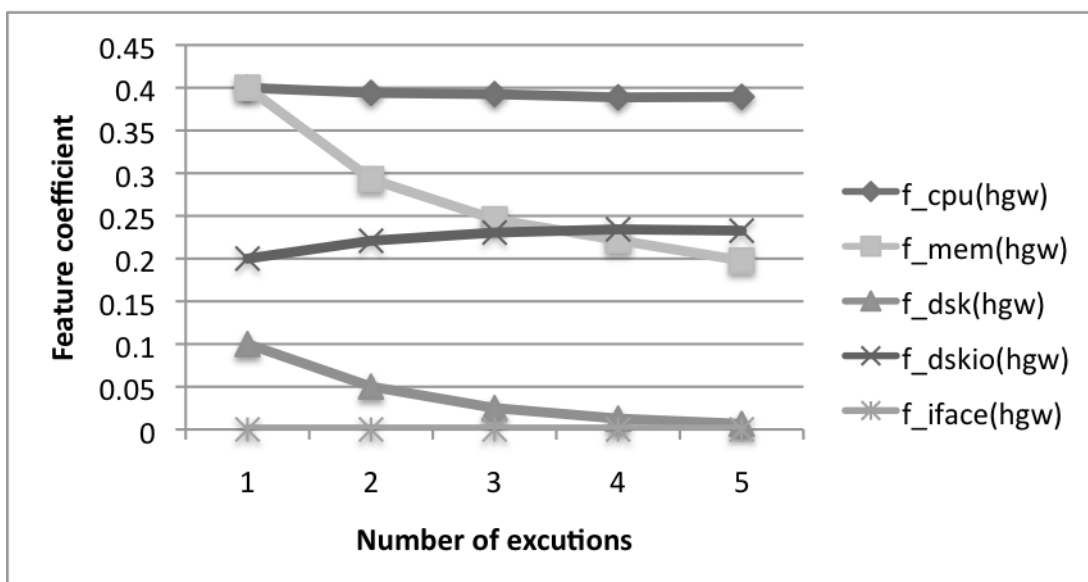


図 5.5: 実験 1 の特徴係数の変化 (HGW 実行時)

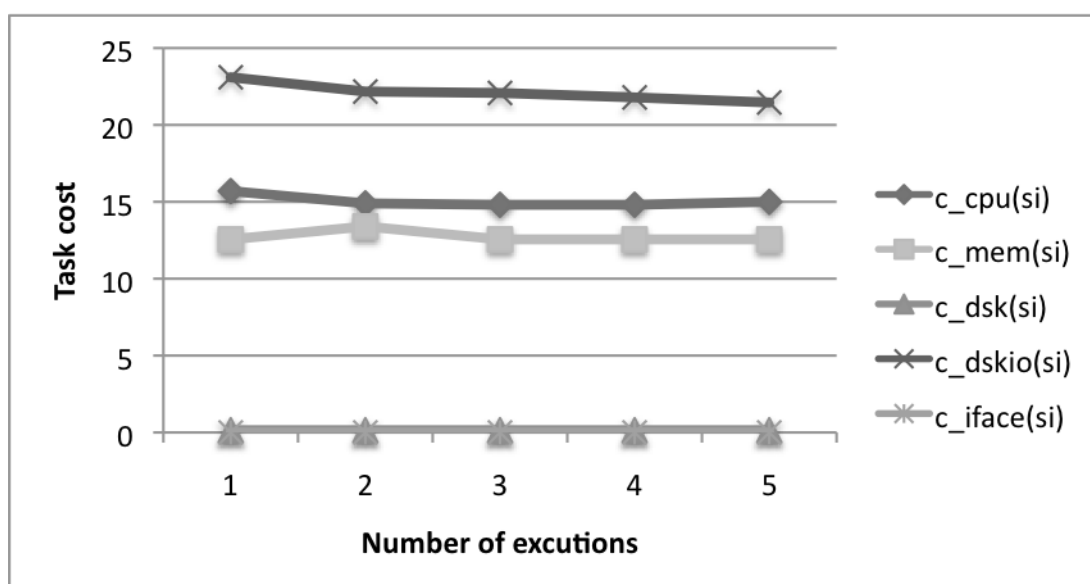


図 5.6: 実験1 のタスクコストの変化 (SI 実行時)

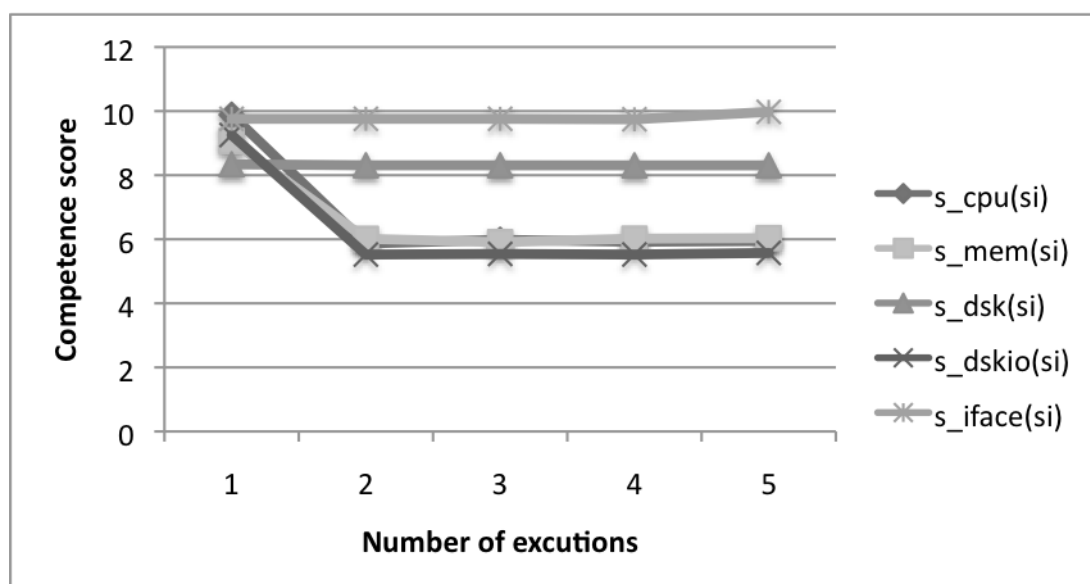


図 5.7: 実験1 の適性スコアの変化 (SI 実行時)

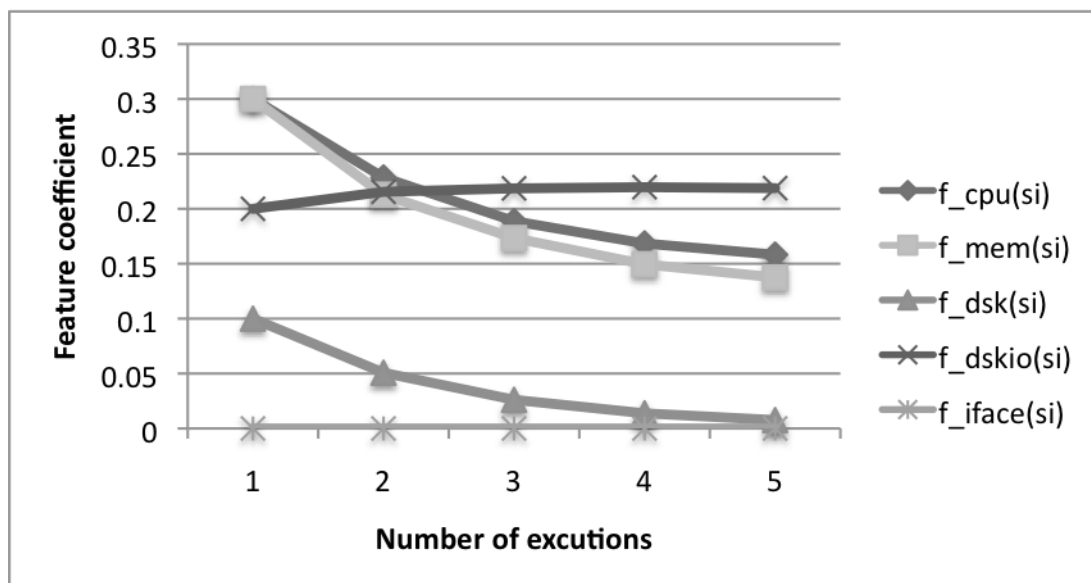


図 5.8: 実験 1 の特徴係数の変化 (SI 実行時)

また，実験 1 において各要素でタスクを実行した場合の実行時間の平均値を表 5.10 に示す．

表 5.10: 実験 1 における実行時間の平均値

	実行時間 [sec]
HGW	130.34
SI	77.89

実行時間を比較すると，SI の方が HGW よりも短い実行時間で終了していることが分かる．また， s_r^{si} は s_r^{hgw} より高い適性スコアを持っている数が多い．これより，適性スコアの高い要素でタスク τ_{anlz} を実行した方が低い要素で実行する場合と比べて，実行時間も短くなることが分かった．このため，提案手法が有効性は高いことがいえる．

5.4.2 複数タスクの配置先決定実験結果

次に複数のタスクを配置する場合の実験結果について説明する．実験2も同様に，初回実行時に計測された資源使用率，計算された適性スコア，タスク実行後に計算されたタスクコスト，更新された特徴係数を示す．

まず，タスク実行前の HGW 及び SI の資源使用率を以下に示す，

表 5.11: 実験2における両タスクの実行前に計測された資源使用率

	$u_{r_{cpu}}^e$	$u_{r_{mem}}^e$	$u_{r_{dsk}}^e$	$u_{r_{dskio}}^e$	$u_{r_{iface}}^e$
HGW	1.51	12.1	31.9	3.11	0.00051
SI	0.32	3.07	7.09	0.15	0.004

次に，取得した資源使用率（表 5.11）の値を基に計算される，解析タスク τ_{anlz} を配置する際に計算した適性スコアを表 5.12 に示す．

表 5.12: 実験2における解析タスク τ_{anlz} の配置時に計算された適性スコア

	$s_{\tau_i, r_{cpu}}^e$	$s_{\tau_i, r_{mem}}^e$	$s_{\tau_i, r_{dsk}}^e$	$s_{\tau_i, r_{dskio}}^e$	$s_{\tau_i, r_{iface}}^e$
HGW	9.66	7.57	4.79	9.31	10.00
SI	9.90	9.15	8.34	9.96	10.00

次に，解析タスク τ_{anlz} の実行後に計測したタスクコストを表 5.13 に示す．

表 5.13: 実験2における解析タスク τ_{anlz} の初回実行時に計測されたタスクコスト

	$c_{r_{cpu}}^e$	$c_{r_{mem}}^e$	$c_{r_{dsk}}^e$	$c_{r_{dskio}}^e$	$c_{r_{iface}}^e$
HGW	44.92	23.14	0.0239	18.0	0
SI	19.0	18.9	0.150	28.1	0

次に，タスクコストを基に再計算された特徴係数を表 5.14 に示す．

以下に，HGW で繰り返し実行した場合に計測したタスクコスト，適性スコア，特徴係数の変化を，図 5.9 から図 5.11 に示す．

図 5.9 は，HGW 上でタスク τ_{anlz} を実行する毎に計測したタスクコストの変化の様子を示している．図 5.11 は，HGW 上でタスク τ_{anlz} を実行する前に計算された適性スコアの変化の様子を示している．図 5.11 は，HGW 上でタスク τ_{anlz} を実行する毎に計算されたタスクコストを基に計算された特徴係数の変化の様子を示している．

表 5.14: 実験2におけるタスクコスト算出後に再計算された特徴係数

	$f_{r_{cpu}}^e$	$f_{r_{mem}}^e$	$s_{r_{dsk}}^e$	$f_{r_{dskio}}^e$	$f_{r_{iface}}^e$
HGW	0.425	0.316	0.0501	0.190	0
SI	0.245	0.244	0.0507	0.241	0

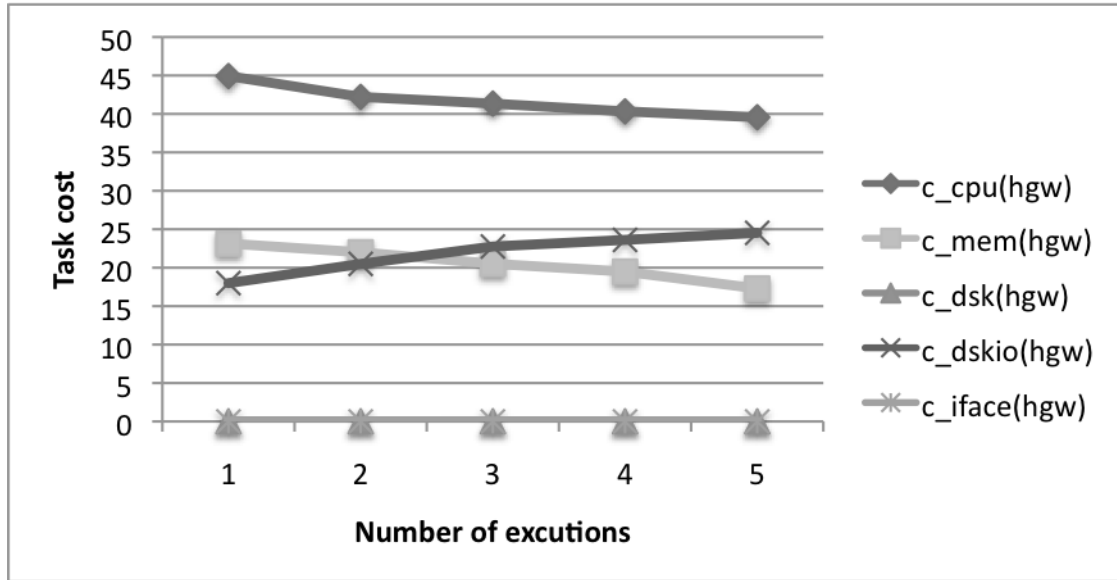


図 5.9: 実験2のタスクコストの変化 (HGW 実行時)

次に、SIで τ_{anlz} を繰り返し実行した場合に計測したタスクコスト、適性スコア、特徴係数の変化を、図5.12から図5.14に示す。

図5.12は、SI上でタスク τ_{anlz} を実行する毎に計測したタスクコストの変化の様子を示している。図5.14は、SI上でタスク τ_{anlz} を実行する前に計算された適性スコアの変化の様子を示している。図5.14は、SI上でタスク τ_{anlz} を実行する毎に計算されたタスクコストを基に計算された特徴係数の変化の様子を示している。

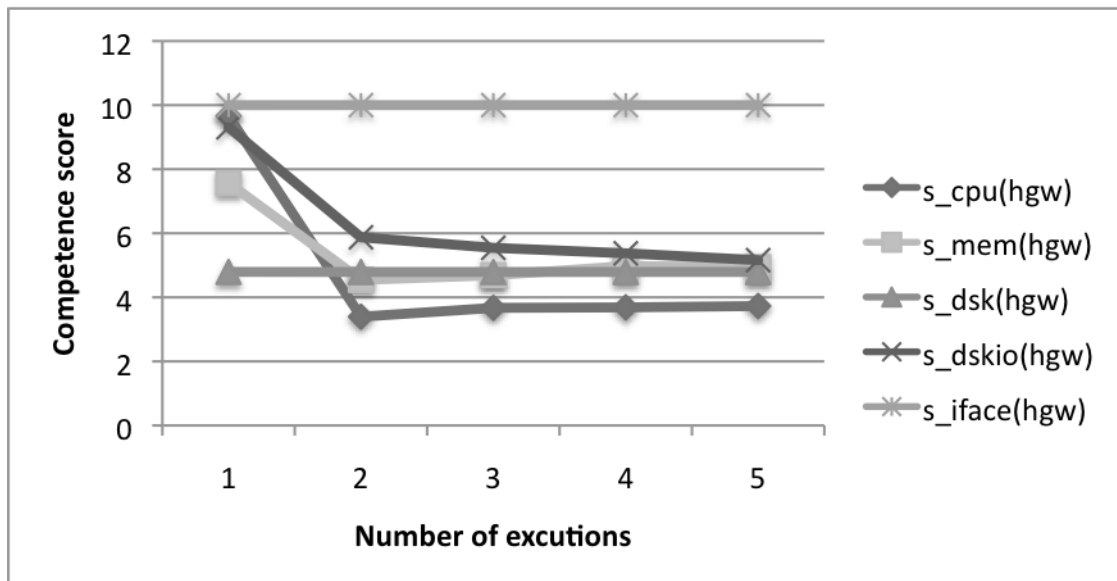


図 5.10: 実験 2 の適性スコアの変化 (HGW 実行時)

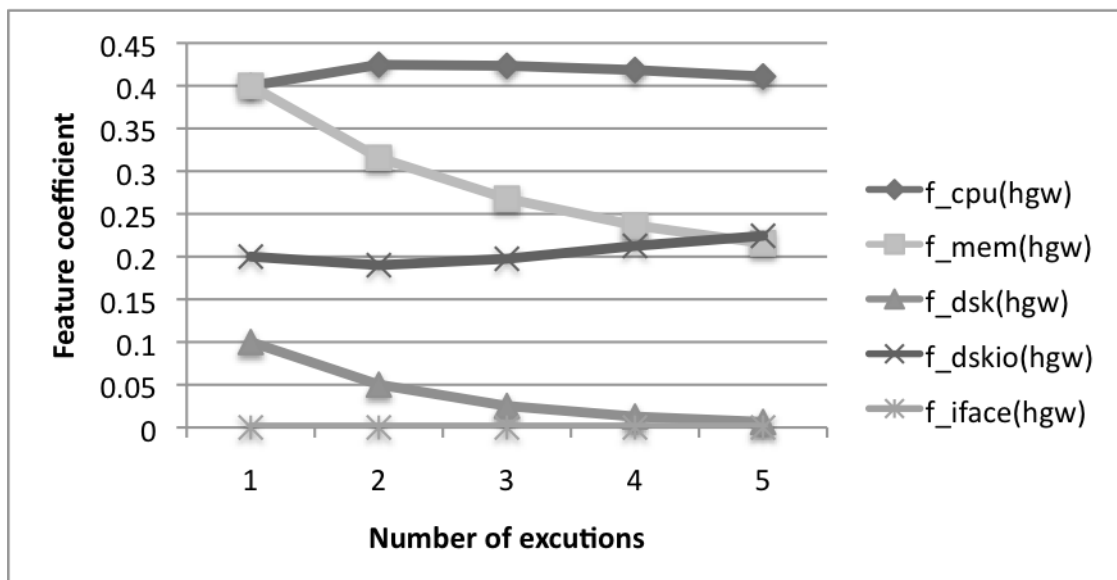


図 5.11: 実験 2 の特徴係数の変化 (HGW 実行時)

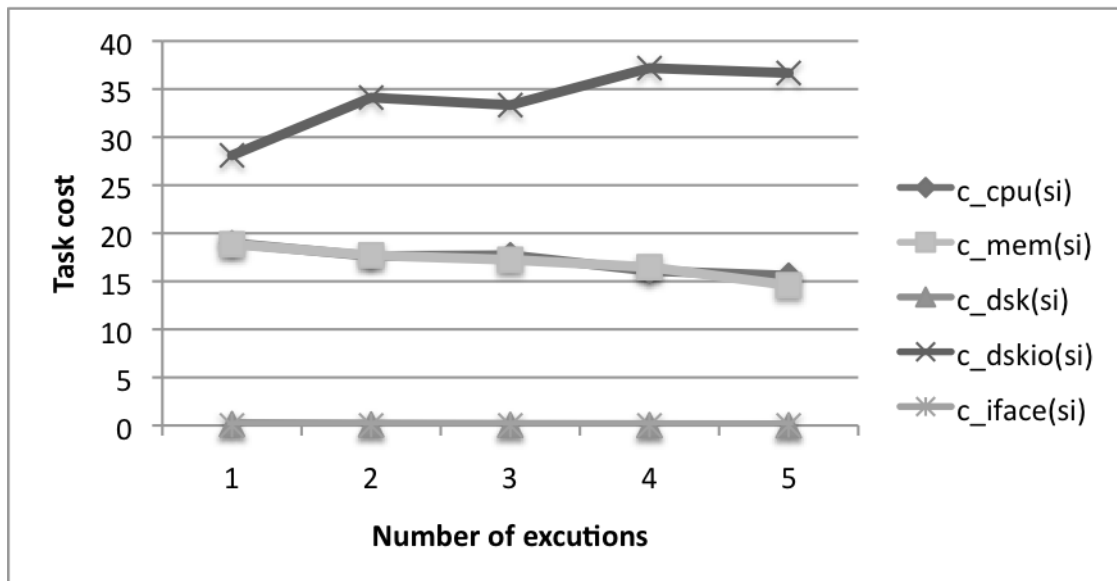


図 5.12: 実験 2 のタスクコストの変化 (SI 実行時)

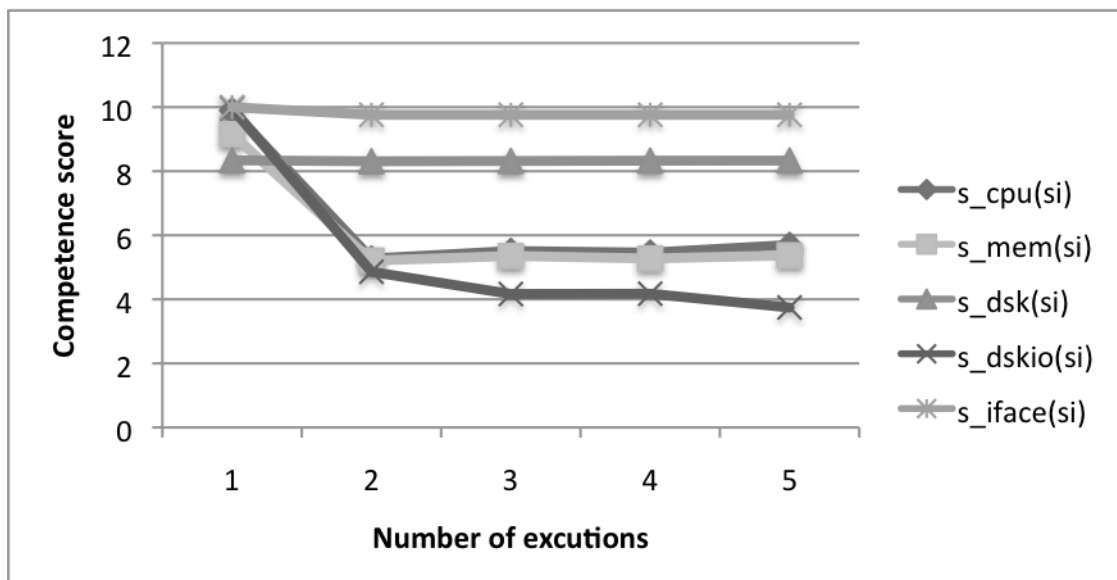


図 5.13: 実験 2 の適性スコアの変化 (SI 実行時)

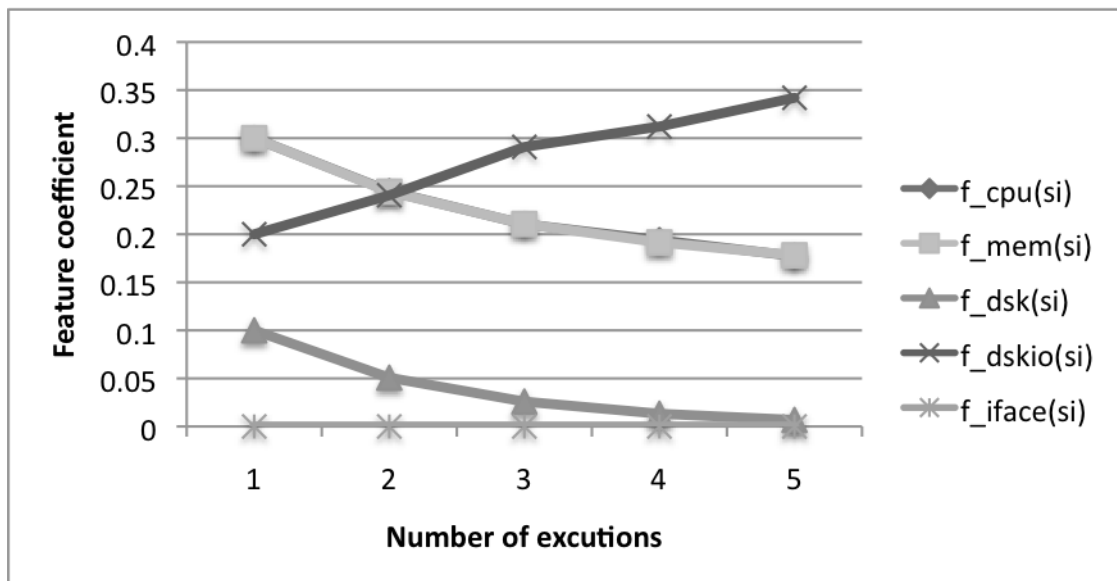


図 5.14: 実験 2 の特徴係数の変化 (SI 実行時)

また，実験 2 において各要素でタスクを実行した場合の実行時間の平均値を表 5.15 に示す．

表 5.15: 実験 2 における実行時間の平均値

	応答時間 [sec]
HGW	230.1
SI	322.9

この場合も同様に， s_r^{si} は s_r^{hgw} より高い適性スコアを持っている数が多い．しかし，先ほどの実験とは異なり，HGW で全てを実行する場合の方が応答時間は短い結果となった．

第6章 考察

考察では，本研究で提案した適性スコア，タスクコスト，特徴係数の有効性について実験結果を基に検証する．6.1 節ではタスクコストと特徴係数を，6.2 節では適性スコアの有効性について検証する．また6.3 節では，一般的なホームネットワークを利用したサービスへの適用性について検証する．

6.1 タスクコストと特徴係数の有効性

本節では本研究で提案したタスクコスト及び特徴係数の有効性について，実験結果を基に検証する．

まず，タスクコストの有効性について検証する．タスクコストは，タスクが未実行のときと実行中のときの資源使用率の変化量をパーセントポイントで表したものである．このため，タスクの実行毎にタスクコストの値は更新される．タスクコストはタスクが実行終了後に特徴係数と過去に計算されたタスクコストを基に再計算される．

実験1において，図5.4ではHGWで，図5.7ではSIで，継続的にタスクを実行した場合のタスクコストの推移を表したものである．実験1で求められたタスクコストの平均値を表6.1に示す．

表 6.1: 実験1で求められたタスクコストの平均値

	c_{cpu}^e	c_{mem}^e	c_{dsk}^e	c_{dskio}^e	c_{iface}^e
HGW	38.6	18.8	0.029	23.37	0.001
SI	15.0	12.7	0.15	22.1	0.001

実験1では，同時に実行されるタスクが一つのみであるため，実行前の平常時の資源使用率と実行中のタスク実行時の資源使用率との差分を利用する．このため，タスク実行終了直後に計測した資源使用率と直前の実行時に計測した資源使用率との差分がそのままタスクコストとなる（式(4.9)）．HGWで実行した場合の方がSIで実行した場合と比べ，ほとんどの資源でのタスクコストは高い値となるが，これはHGWの持つ資源の大きさがSIの持つ資源の大きさと比較して少ないためである．また， u_{dskio} はHGWとSIもほぼ

同じ結果となったが、これは HGW の持つ資源 r_{diskio} と SI の持つ資源 r_{diskio} の大きさが同等であったためと考えられる。

次に実験2のタスクコストの変化について説明する。実験2において、図5.10には HGW で、図5.13には SI で継続的にタスクを実行した場合のタスクコストの推移を表したものである。実験2で求められたタスクコストの平均値を表6.2に示す。

表 6.2: 実験2で求められたタスクコストの平均値

	$c_{r_{cpu}}^e$	$c_{r_{mem}}^e$	$c_{r_{dsk}}^e$	$c_{r_{dskio}}^e$	$c_{r_{iface}}^e$
HGW	41.7	20.5	0.021	24.28	0
SI	17.2	18.3	0.075	35.5	0

実験2では同時に実行されるタスクが複数あるため、タスクコストは直前の実行時に計測した資源使用率の差分に、全体の特徴係数の和の中でタスクコストを求めたいタスクが持つ特徴係数の割合を掛けた値と、過去に計測したタスクコストとの平均を取ることで求められる（式(4.9)）。実験1と同じ解析タスク τ_{anlz} のタスクコストを求めたが、実験1の結果と比較すると、異なる値が計算されたことが分かる。

表6.1と表6.2より両要素のタスクコストを比較すると、いくつかの資源 ($r_{cpu}, r_{mem}, r_{diskio}$) において実験1で求めたタスクコストより実験2で求めたタスクコストの方が高くなっていることが分かる。これは、特徴係数の初期値として決定した値が不適切であったか、または実際にタスクを実行する際に消費する資源の割合が増加したと考えられる。

特徴係数の実行結果は図5.5、図5.8、図5.11、図5.14に示したが、比較を行うため実験1における両要素上で計算された特徴係数の変化の様子を図6.1に、実験2における両要素上で計算された特徴係数の変化の様子を図6.2にそれぞれ示す。ここで示す資源は r_{cpu}, r_{mem} とした。

両実験で求めた特徴係数の変化を見ると、初期値と二回目以降は異なる値に推移していることが分かる。特徴係数はタスクを実行することで求められるタスクコストの割合の値に近い値に計算されるため、特徴係数の初期として実際のタスクコストから離れている値を与えてしまうと、再計算で適した値に近づくまでに時間が必要となる。

図6.1と図6.2を見ると、どちらの要素上でも再計算を5回繰り返すことでタスクコストにおけるパーセント表記での値に近くなることが分かる。この結果、どの実験のどちらの要素でも同様に計算を5回繰り返すことでタスクコストの値と近くなることが分かる。

結論として、特徴係数の初期値を正しく設定しないとタスクコストも正確な値が求められないことが分かった。正確なタスクコストが求められないと、正確な適性スコアが計算できなくなってしまうことは適性スコアの導出式(4.6)からも明らかである。しかし、実行終了後にタスクコストを基に再計算することで特徴係数は正確な値に近づくため、導出されるタスクコストは正確な値に近づくことが分かった。よって、提案した再計算手法は適切であったと言える。

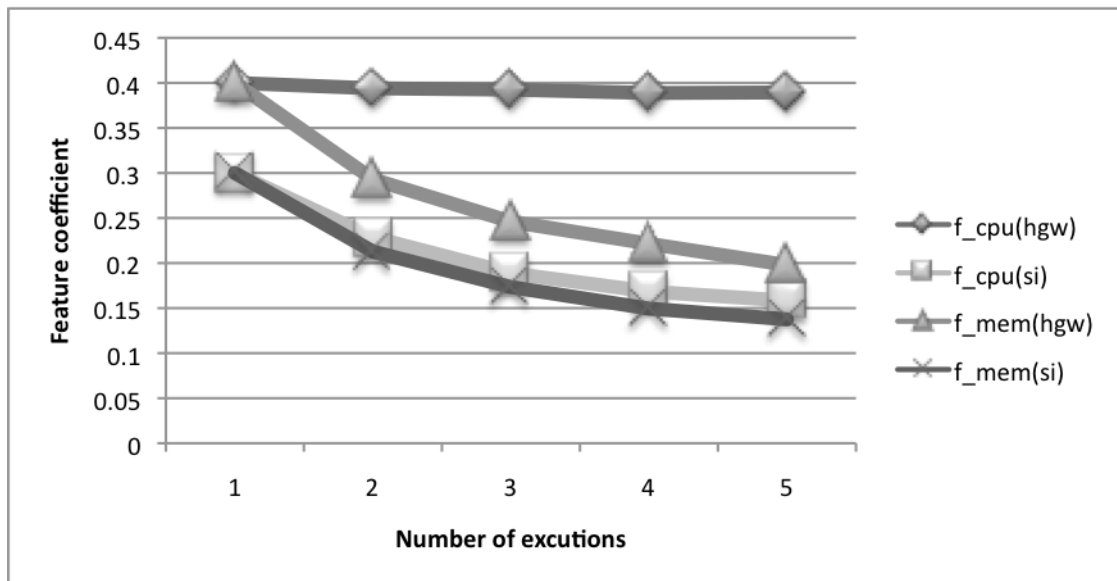


図 6.1: 実験 1 の特徴係数の比較 (r_{cpu}, r_{mem})

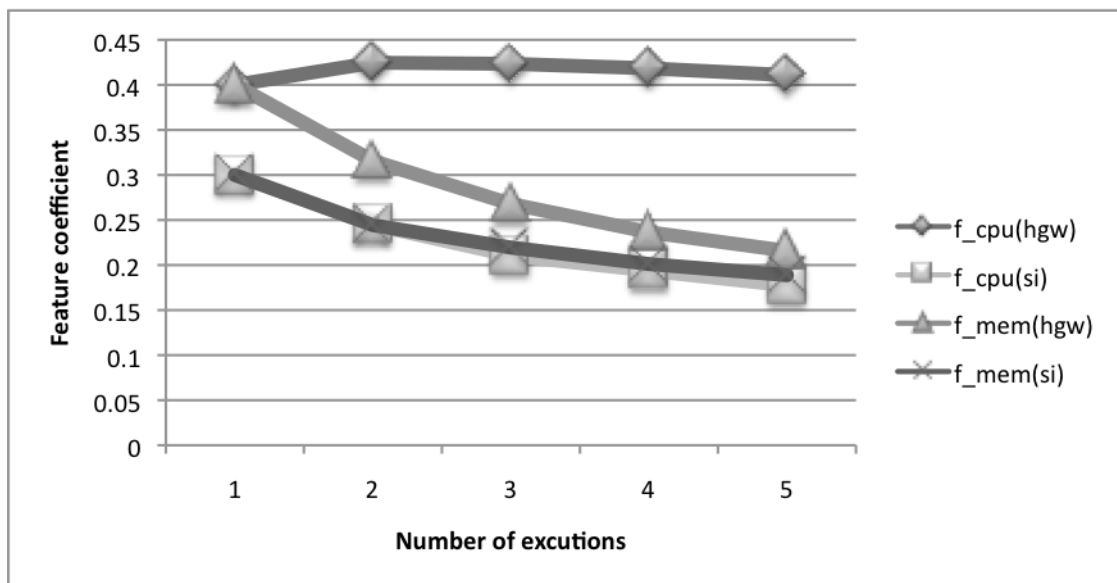


図 6.2: 実験 2 の特徴係数の比較 (r_{cpu}, r_{mem})

6.2 適性スコアの有効性

適性スコアは、実行したいタスクがどれだけ適しているかを表わしており、資源使用率やタスクコストを基に計算される値である。

適性スコアの実験結果は図 5.4, 5.7, 5.10, 5.13 に示しているが、本節で比較を行うため実験 1 における両要素上で計算された適性スコアの変化の様子を図 6.3 に、実験 2 における両要素上で計算された適性スコアの変化の様子を図 6.4 にそれぞれ示す。

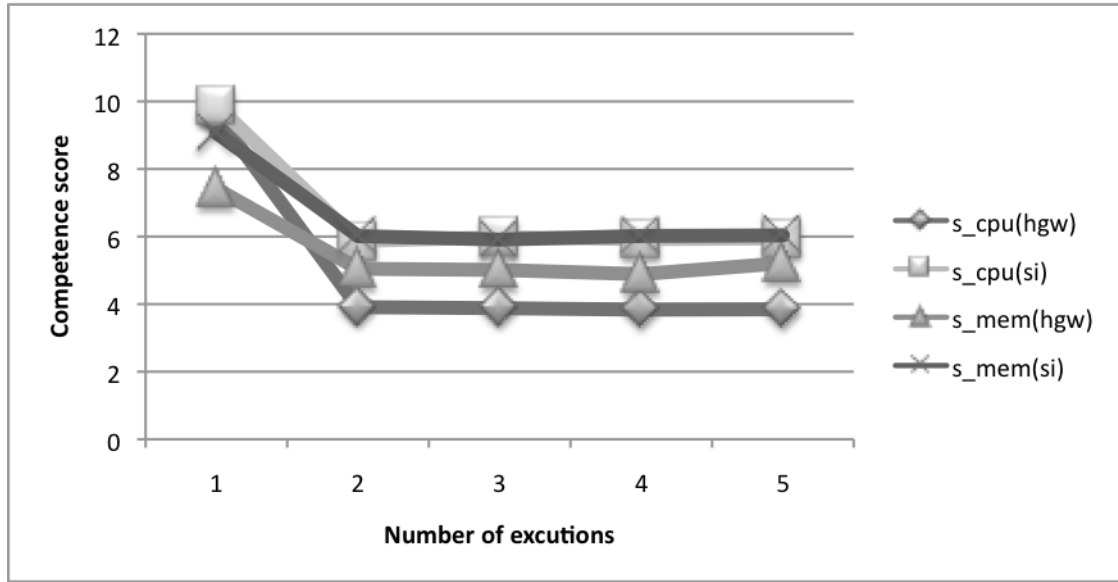


図 6.3: 実験 1 の適性スコアの比較 (r_{cpu}, r_{mem})

図 6.3 と図 6.4 において、両実験とも一回目の適性スコアだけ値が大きく異なる資源が存在する場合がある。これは一回目の適性スコアの計算時のタスクコストには初期値が入るためである。タスクコストの初期値は 0 となるため、適性スコアの導出式 (4.6) より適性スコアは高い値を示すことが分かる。

また資源使用率の高い場合、適性スコアが低くなることが適性スコアの導出式 (4.6) から分かる。実験 1 において、表 5.6 に示した取得した資源使用率を要素毎に比較すると、 r_{iface} 以外の全ての資源使用率は SI より HGW の方が高い状態であることが分かる。これより、両要素上の資源のうち $r_{cpu}, r_{mem}, r_{dsk}$ の適性スコアは HGW より SI の方が高い値を示すことが推測される。

実際に計算された適性スコアを表 5.7 に示した結果より比較すると、 $r_{cpu}, r_{mem}, r_{dsk}$ の適性スコアは HGW より SI の方が高い値を示すことが分かった。しかし、 r_{dskio} と r_{iface} の適性スコアは、SI よりも HGW の方が高い値を示していることが分かる。各パラメータを比較すると、この理由は設定した資源使用率上限値が l_r^{si} は l_r^{hgw} より低いためである。つまり、SI と HGW の資源使用率の差が小さい場合、 s_r^{si} より s_r^{hgw} の方が高い値を取るよ

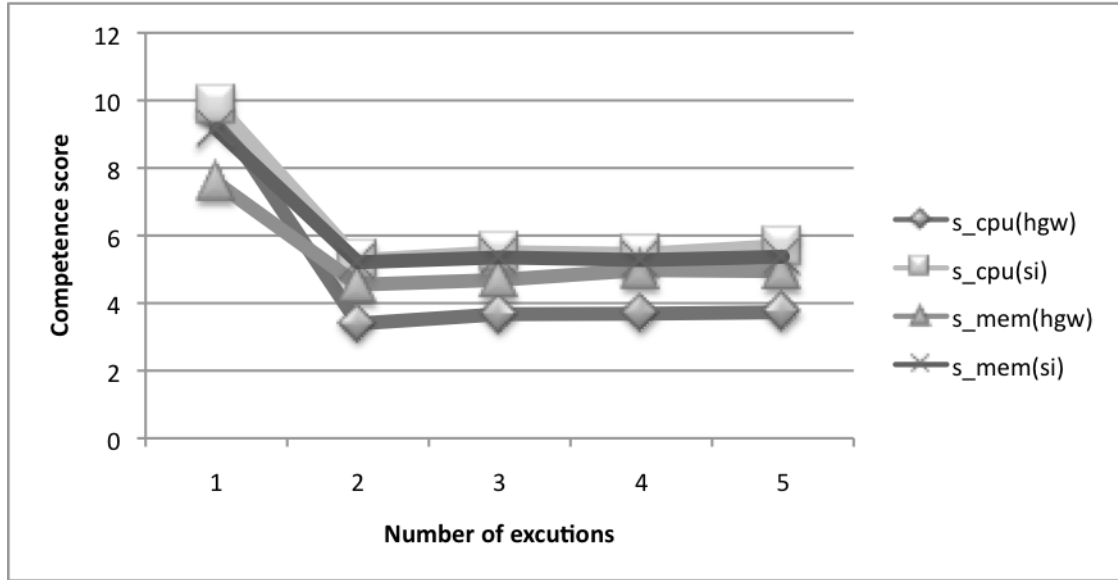


図 6.4: 実験 2 の適性スコアの比較 (r_{cpu}, r_{mem})

うになり、SI より HGW を選択されることが分かった。つまり、HGW の負担が SI より小さい場合と同程度の場合は HGW へ、HGW にかかる負荷の方がある程度大きい場合は SI へ配置先が決定されるという、意図した挙動が実現されていることが確認できた

次に、適性スコアがどれくらい有効であるか考察するため、タスクの実行時間との関係について比較する。

6.2.1 適性スコアとタスクの実行時間の関係

適性スコアとタスクの実行時間の比較を行う。表 5.10 は実験 1 において、表 5.15 は実験 2 において計測されたタスク実行時間の平均値を示す。

実験 1 では、どの実行時でも $S_{\tau_{anlz},r}^{si}$ は $S_{\tau_{anlz},r}^{hgw}$ より高い値を持つ資源が多いため、HGW より SI を選択するようになっている。実行時間を比較しても HGW での実行時間より SI での実行時間の方が短いことが分かり、SI で実行した方が有効であることが言える。そのため、実験 1 においては適性スコアが実行時間の観点からも有効性の高い提案手法であることが分かった。

実験 2 も実験 1 と同様で、どの実行時でも $S_{\tau_{anlz},r}^{si}$ は $S_{\tau_{anlz},r}^{hgw}$ より高い値を持つ資源が多いため、HGW より SI を選択するようになっている。しかし、実行時間を比較すると、HGW より SI の方が実行時間が長くなることが分かった。実験 1 より、解析タスク τ_{anlz} 単体の実行時間は HGW より SI の方が短いため、通信タスク τ_{comm} の実行時間が影響したものと考えられる。通信タスクのタスクコスト $c_{\tau_{comm},r}^e$ の資源毎の平均値を表 6.3 に示す。

表 6.3 より、HGW で計測されたタスクコスト $c_{\tau_{comm},r}^{hgw}$ より SI で計測されたタスクコス

表 6.3: 実験別に求められた通信タスク τ_{comm} のタスクコスト平均値の比較

	$c_{\tau_{comm}, r_{cpu}}^e$	$c_{\tau_{comm}, r_{mem}}^e$	$c_{\tau_{anlz}, r_{dsk}}^e$	$c_{\tau_{comm}, r_{dskio}}^e$	$c_{\tau_{comm}, r_{iface}}^e$
HGW	20.9	14.8	0.071	12.03	0.056
SI	14.3	15.4	0.23	10.8	41.6

ト $c_{\tau_{comm}, r}^{si}$ の方が高い資源が多いことが分かる。よって、通信タスク τ_{comm} の影響により、実行時間も SI の方が長くなることが推測できる。このため、実験 2 において解析タスク τ_{anlz} の配置先を決定する際には提案手法を利用することで負荷を考慮した配置先の決定は可能だが、実行時間を考慮すると同時に実行される通信タスク τ_{comm} の影響も考慮した配置先の決定が重要となる。

6.3 一般的なサービスでの適用性

6.3.1 一般的なサービスの紹介

現在，実在するホームネットワークを利用したサービスは少なく，一般的なサービスへの適用性を議論することは困難である．このため，現在考えられているサービスについて調査し [14]，そのサービスへの適用性について議論した．

一般的なホームネットワークを利用したサービスとして9つの例について説明する．

- ユビキタスな安心・安全サービス
- マルチメディアな安心・安全サービス
- ユビキタスな教養・娯楽サービス
- 健康・長寿なヘルスケアサービス
- 省エネ・エコなホームマネジメントサービス
- 快適・ふれあいコミュニケーションサービス
- 高度な介護サービス利用と負担の軽減
- 家電などの故障対応コールセンター
- 緊急地震速報との連携による地震初期対応の自動化

ユビキタスな安心・安全サービス

ユビキタスな安心・安全サービスでは，宅内や地域社会において監視・介護サービスを実現する．宅内に設置されたカメラやセンサから宅内の情報を取得し，宅外からカメラ映像の確認や警備会社などによる宅内状態の監視を行うことが可能である．このサービスの概要を図 6.5 に示す．

SI モデルを利用した場合，サービスは以下の手順で実行される．

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（宅内情報の圧縮）を配布し，SI はタスク 3（宅内情報の解析）とタスク 4（宅内異常の通知）を配置

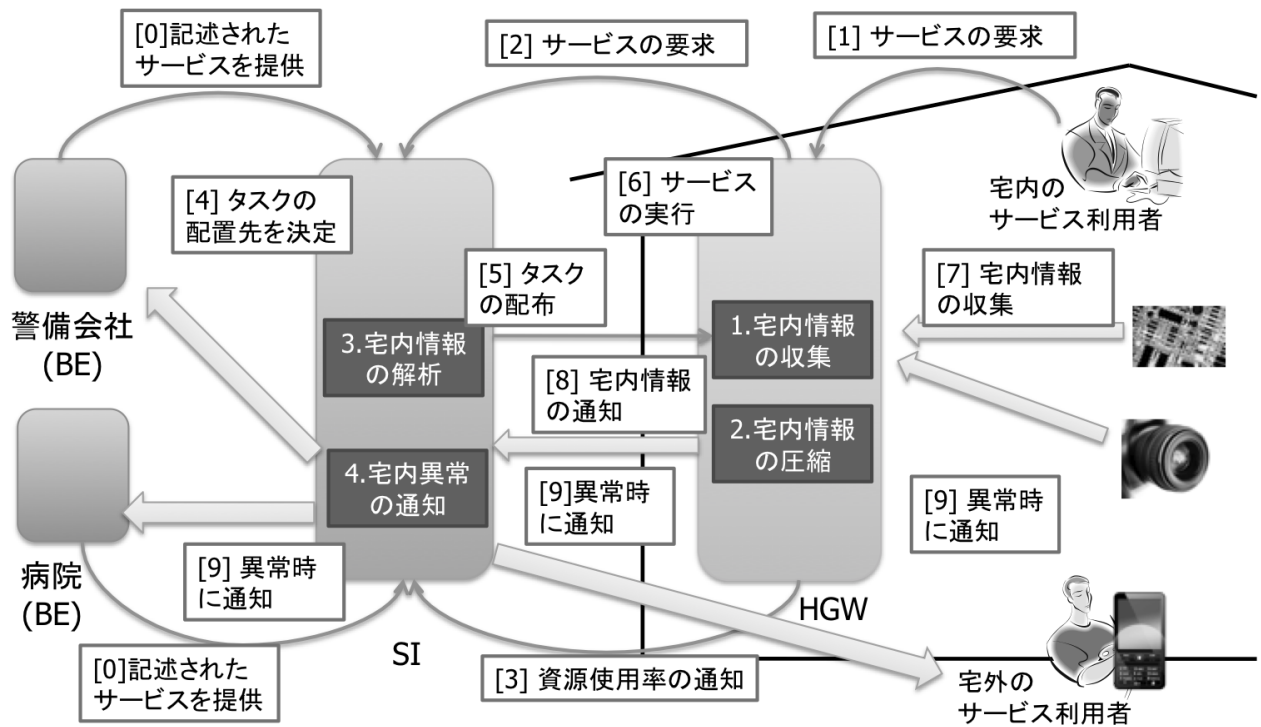


図 6.5: ユビキタスな安心・安全サービス

6. タスクが実行されサービスが開始

7. HGW 上のタスク 1 は、宅内の機器（カメラやセンサなど）からデータを収集

8. HGW 上のタスク 2 は収集したデータを圧縮し、HGW から SI へ送信

9. SI 上のタスク 3 は収集したデータを解析し情報を抽出、異常を検知した場合はタスク 4 が警備会社 (BE) や病院 (BE)、宅内機器へ通知

また、このサービスを実現するために以下のタスクが配置され、実行される。

タスク 1 宅内の機器からデータを収集

タスク 2 宅内で収集したデータを圧縮

タスク 3 宅内情報を解析してサービス利用者や宅内の状態に関する情報を抽出

タスク 4 外部の BE やサービス利用者に対して情報を通知

マルチメディアな安心・安全サービス

マルチメディアな安心・安全サービスは，訪問者への対応や防犯対策，防災対策を実現する．宅内に設置されたカメラの映像を外部から視聴することにより訪問者への対応が可能となったり，警備会社から侵入者の監視を行うことなどのサービスが実現可能となる．このサービスの概要を図 6.6 に示す．

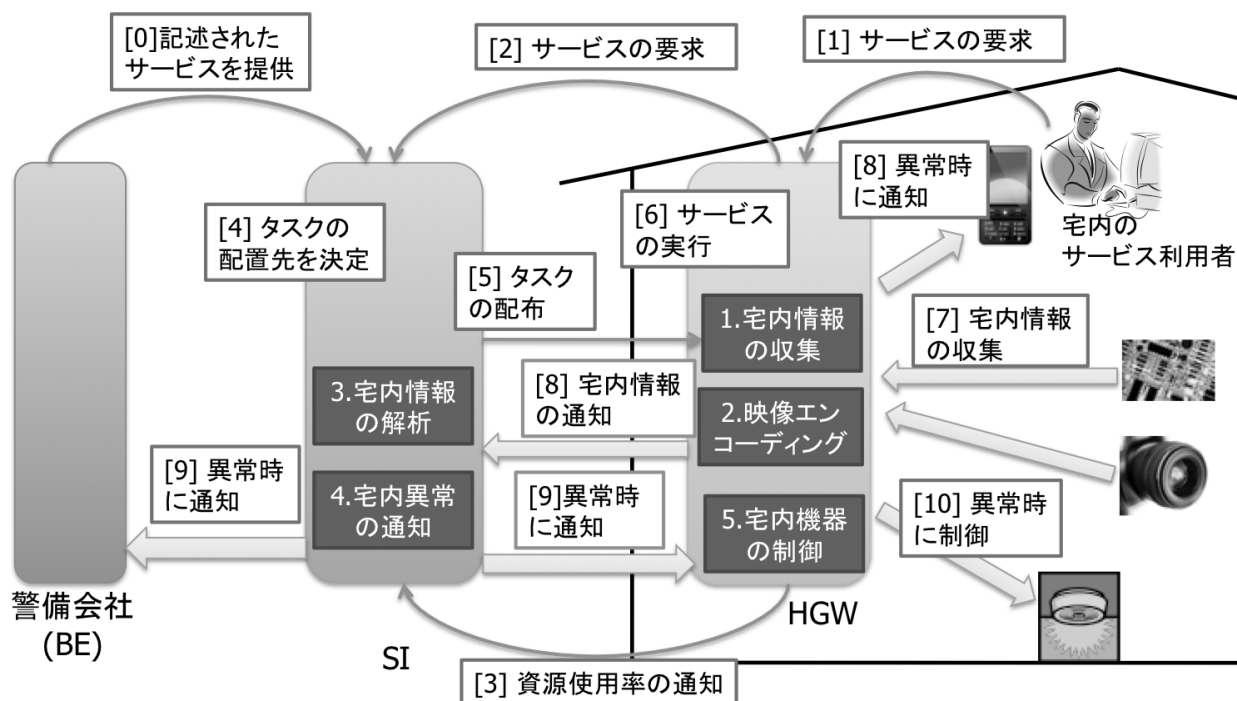


図 6.6: マルチメディアな安心・安全サービス

SI モデルを利用した場合，サービスは以下の手順で実行される．

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（映像エンコーディング），タスク 5（宅内機器の制御）を配布し，SI にタスク 3（宅内情報の解析）とタスク 4（宅内異常の通知）を配置
6. タスクが実行されサービスが開始

7. HGW 上のタスク 1 は、宅内の機器（カメラやセンサなど）からデータや映像を収集
8. HGW 上のタスク 2 は収集した映像をエンコードし、HGW が収集したそれらのデータを SI へ送信
9. SI 上のタスク 3 は収集したデータを解析し情報を抽出、異常を検知した場合はタスク 4 が警備会社や病院 (BE)、宅内機器へ通知
10. HGW 上のタスク 5（宅内機器の制御）から家電機器や住宅設備などを制御

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 宅内の機器からデータを収集

タスク 2 映像のエンコーディング

タスク 3 宅内情報の解析

タスク 4 外部へ宅内情報を送信

タスク 5 家電機器や住宅設備などの宅内に設置された機器を制御

ユビキタスな教養・娯楽サービス

ユビキタスな教養・娯楽サービスは、映像や音声などのコンテンツを、宅内に限らず宅外でも視聴可能とするサービスである。視聴する端末もノート PC や携帯電話、PDA などの様々な機器を使用することが可能となる。このサービスの概要を図 6.7 に示す。

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 2（ストリーミング配信）を配布し、SI にタスク 1（映像エンコーディング）とタスク 2（ストリーミング配信）を配置
6. タスクが実行されサービスが開始

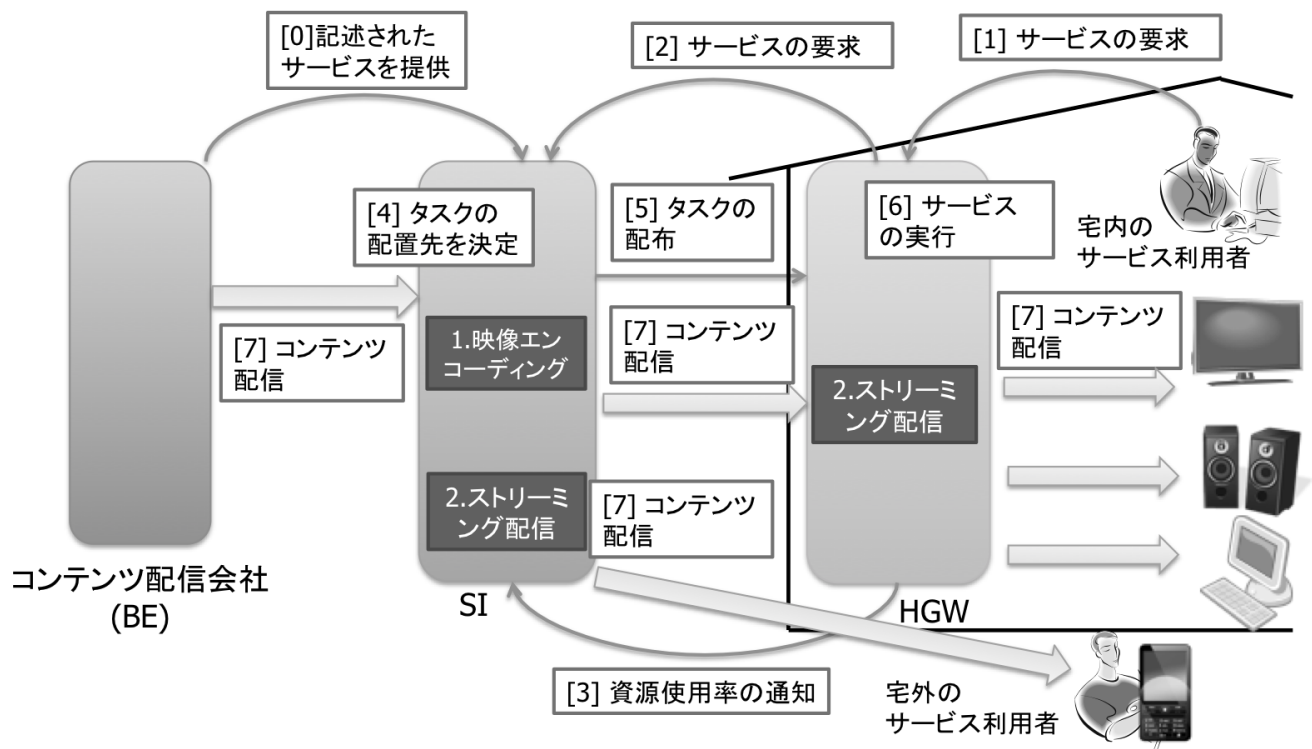


図 6.7: ユビキタスな教養・娯楽サービス

7. BE(コンテンツ配信会社) からコンテンツが配信され、SI 上のタスク 1（映像エンコーディング）がサービス利用者の形態に合わせてコンテンツを変換し、タスク 2（ストリーミング配信）によって宅内や宅外のサービス利用者に配信

また HGW 上のタスク 2（ストリーミング配信）から、宅内に保管されたコンテンツを配信

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 映像のエンコーディング

タスク 2 映像のストリーミング配信

健康・長寿なヘルスケアサービス

健康・長寿なヘルスケアサービスは、宅内で取得可能な健康データを管理することで医療サービスや介護サービスを実現するためのサービスである。宅内に設置されたカメラの映像や体重計や血圧計といった健康機器の情報が、介護サービス提供者や医療サービス提供者へ通知される。宅内のテレビやPCを利用することで、在宅のまま医者や介護者とコミュニケーションを取ることにも可能となる。このサービスの概要を図 6.8 に示す。

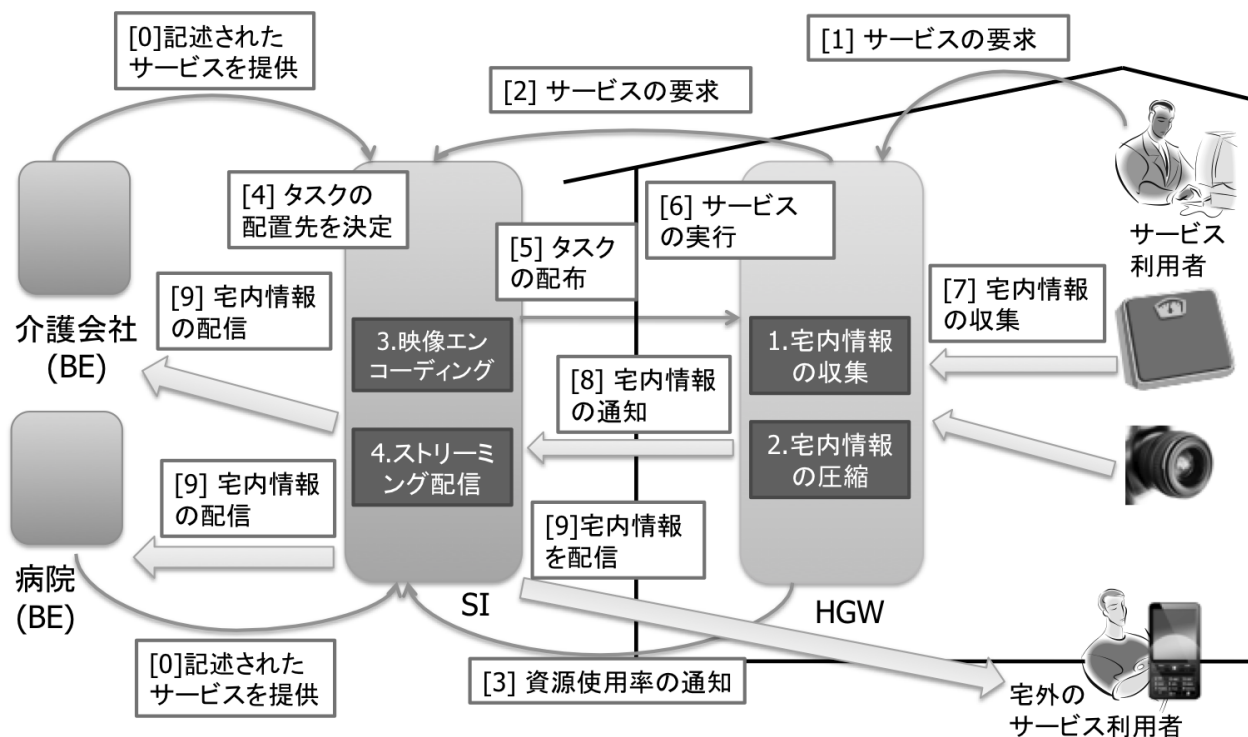


図 6.8: 健康・長寿なヘルスケアサービス

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（宅内情報の圧縮）を配布し、SI にタスク 3（映像エンコーディング）とタスク 4（ストリーミング配信）を配置
6. タスクが実行されサービスが開始
7. HGW 上のタスク 1 は宅内の機器（体重計などの健康機器やカメラなど）からデータや映像を収集
8. HGW 上のタスク 2 は収集した映像を圧縮し、それらのデータを SI へ送信
9. SI 上のタスク 3 は収集した映像をエンコーディングし、タスク 4（ストリーミング配信）によって介護会社 (BE) や病院 (BE)、外部のサービス利用者へ配信

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 宅内の機器からデータを収集

タスク 2 宅内情報を圧縮

タスク 3 映像のエンコーディング

タスク 4 映像のストリーミング配信

省エネ・エコなホームマネジメントサービス

省エネ・エコなホームマネジメントサービスは、宅内で使用する電力量を管理することで省エネルギーを実現する。宅内に設置されたセンサから取得する情報を収集し宅内情報を基に宅内にいるサービス利用者の行動を推測することで宅内の機器の制御を行い、無駄なエネルギー消費を削減することが可能となる。また、過剰な電力消費を行っている場合は電力会社などの外部から省エネルギーの要求が送られ自動的に制御が行われるサービスも考えられている。このサービスの概要を図 6.9 に示す。

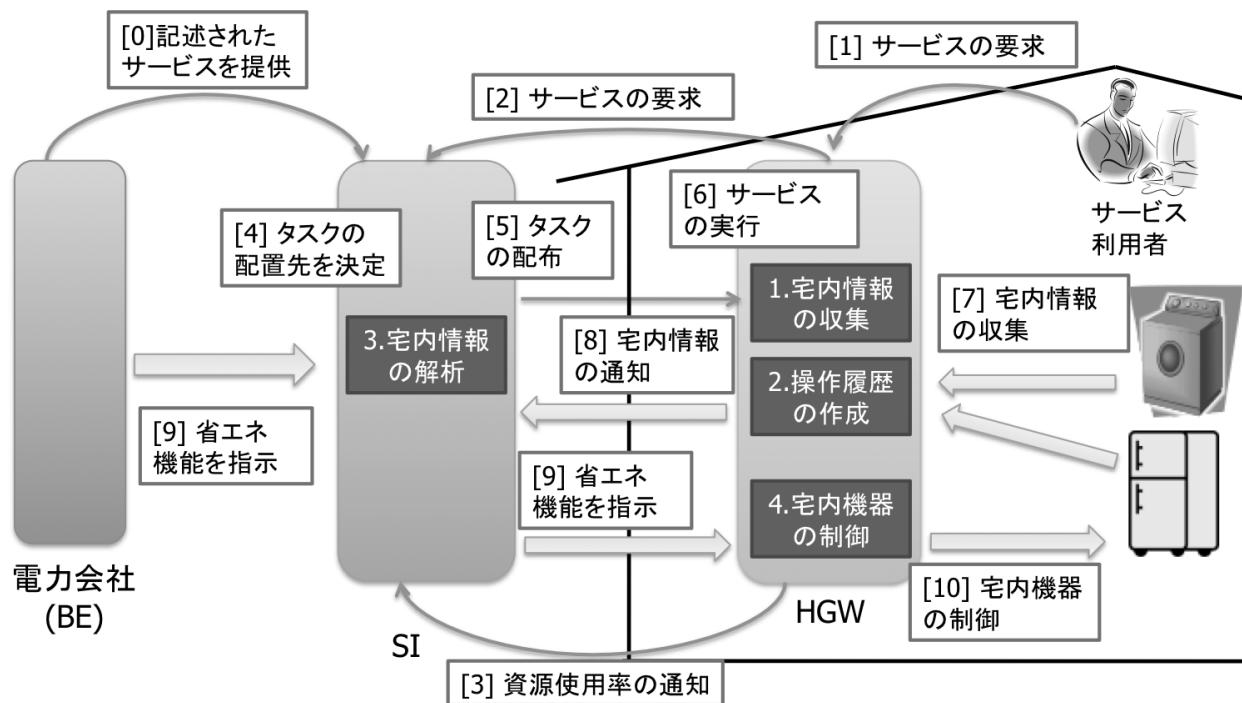


図 6.9: 省エネ・エコなホームマネジメントサービス

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求

2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（操作履歴の作成）、タスク 4（宅内機器の制御）を配布し、SI にタスク 3（住宅情報の解析）を配置
6. タスクが実行されサービスが開始
7. HGW 上のタスク 1 は宅内の機器（家電機器や住宅設備など）から使用状況などのデータを収集し、タスク 2（操作履歴の作成）は収集した情報を基に宅内機器の操作履歴を作成
8. HGW から SI へ宅内情報を通知し、SI 上のタスク 3（宅内情報の解析）では操作履歴から、サービス利用者の宅内機器の使用状況を抽出
9. 電力会社（BE）や外部のサービス利用者から宅内機器の省エネモードを指示する場合、SI から HGW へ省エネモードを通知
10. HGW 上のタスク 4（宅内機器の制御）から家電機器や住宅設備などの宅内機器を制御し省電力を実現

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 宅内の機器からデータを収集

タスク 2 宅内機器から取得したデータを基にサービス利用者の操作履歴を作成

タスク 3 宅内情報を解析してサービス利用者の使用状況に関する情報を抽出

タスク 4 家電機器や住宅設備などの宅内に設置された機器を制御

快適・ふれあいコミュニケーションサービス

快適・ふれあいコミュニケーションサービスは、宅内の機器をコミュニケーション端末として利用するサービスを実現する。PC や携帯電話などの情報端末は操作することが困難なサービス利用者に対して、テレビなどの他の機器を利用したコミュニケーションを実現することが可能となる。このサービスの概要を図 6.10 に示す。

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求

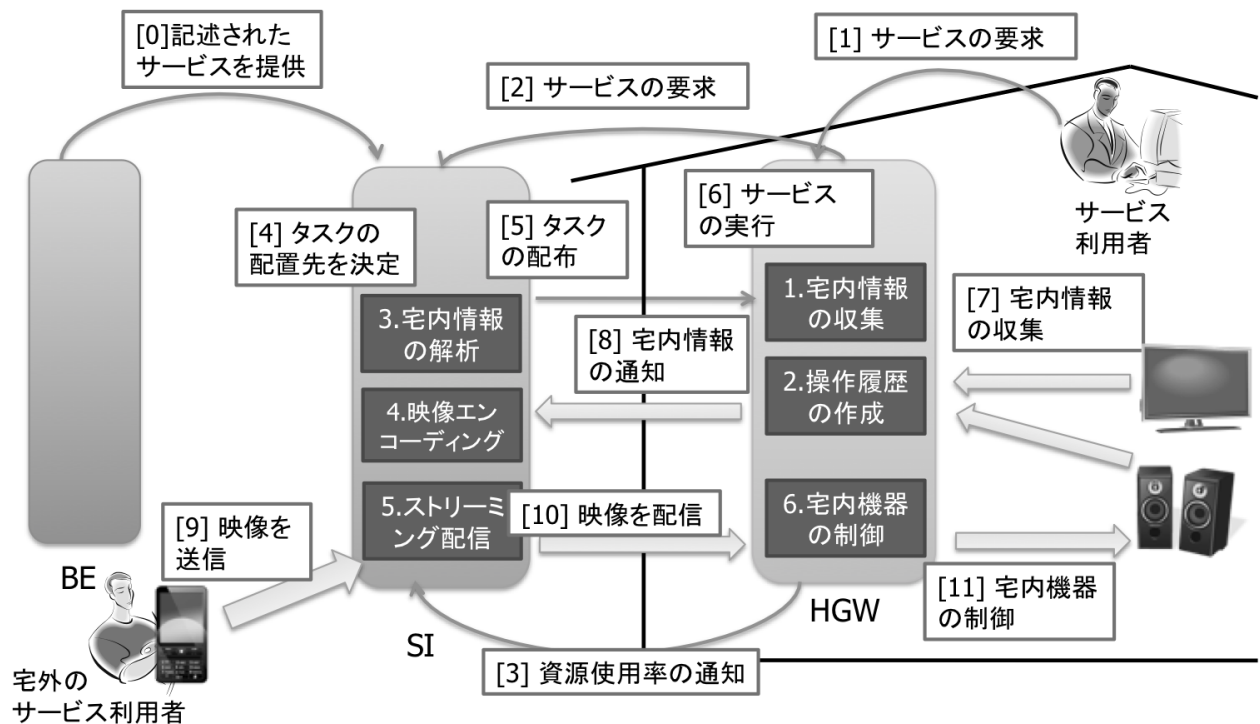


図 6.10: 快適・ふれあいコミュニケーションサービス

2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1 (宅内情報の収集) とタスク 2 (操作履歴の作成), タスク 6 (宅内機器の制御) を配布し, SI にタスク 3 (住宅情報の解析) とタスク 4 (映像エンコーディング), タスク 5 (ストリーミング配信) を配置
6. タスクが実行されサービスが開始
7. HGW 上のタスク 1 は宅内の機器 (家電機器や住宅設備など) から使用状況などのデータを収集し, タスク 2 (操作履歴の作成) は収集した情報を基に宅内機器の操作履歴を作成
8. HGW から SI へ宅内情報を通知し, SI 上のタスク 3 (宅内情報の解析) では操作履歴から, サービス利用者の宅内機器の使用状況を抽出
9. 外部のサービス利用者から映像や音声などのコンテンツを SI へ送信

10. SI 上のタスク 4（映像エンコーディング）は映像をサービス利用者の形態に合わせて変換し、タスク 5（ストリーミング配信）によって配信
11. HGW が SI からコンテンツを受信したら、タスク 6（宅内機器の制御）が使用する宅内機器を制御してコンテンツの再生

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 宅内の機器からデータを収集

タスク 2 宅内機器から取得したデータを基にサービス利用者の操作履歴を作成

タスク 3 宅内情報を解析してサービス利用者の使用状況に関する情報を抽出

タスク 4 映像のエンコーディング

タスク 5 映像のストリーミング配信

タスク 6 家電機器や住宅設備などの宅内に設置された機器を制御

高度な介護サービス利用と負担の軽減

高度な介護サービス利用と負担の軽減のサービスは、宅内の介護情報を基にして介護サービスを実現する。介護情報は宅内に設置されたセンサや家電機器によって取得される情報を基に、サービス利用者の状態を抽出する。サービス提供者は、宅内の状態を基にサービスを提供することが可能となる。例えば、事前に冷蔵庫の中の状態を介護士に伝えておくことで効率的に作業を行うことが可能であったり、在宅したままでもサービス利用者の状態を医師は確認することが可能となり適切な医療処置を行うことが可能である。このサービスの概要を図 6.11 に示す。

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内情報の収集）とタスク 2（宅内情報の圧縮）を配布し、SI にタスク 3（宅内情報の解析）とタスク 4（宅内異常の通知）を配置
6. タスクが実行されサービスが開始

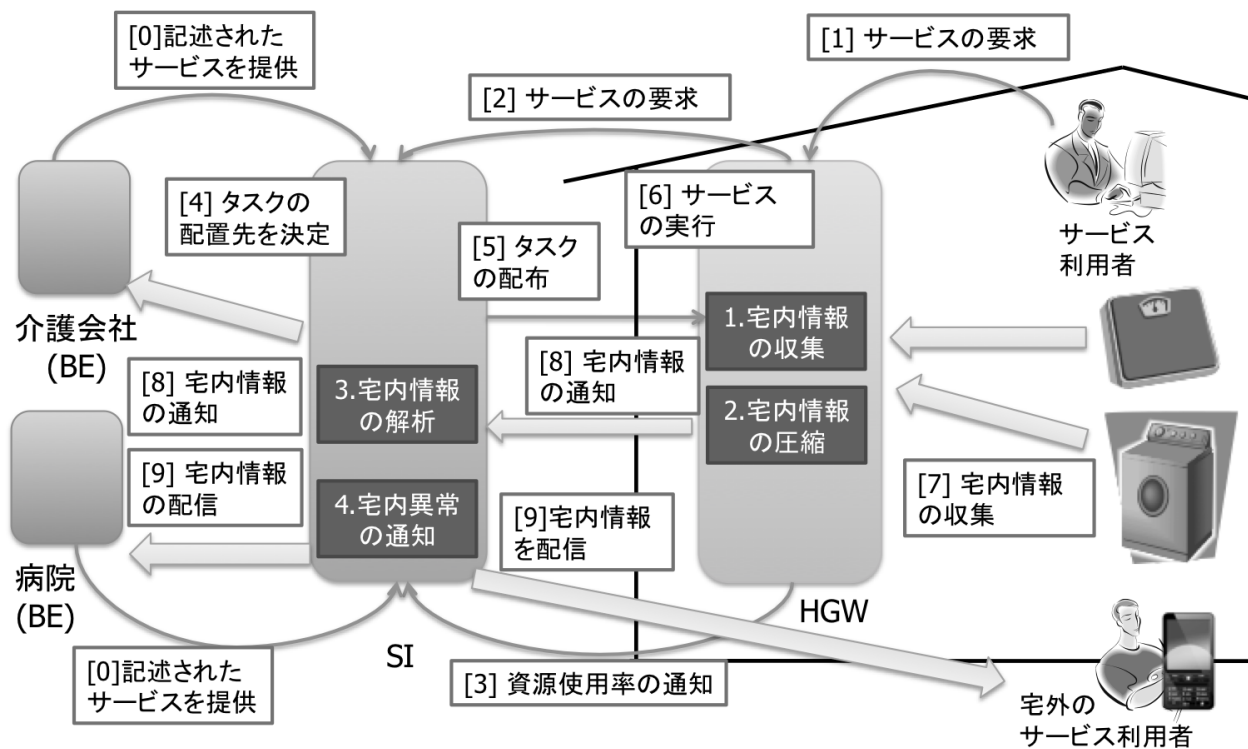


図 6.11: 高度な介護サービス利用と負担の軽減

7. HGW 上のタスク 1（宅内情報の収集）は宅内機器（家電機器や住宅情報など）から情報を収集
8. HGW 上のタスク 2（宅内情報の圧縮）は収集した宅内情報を圧縮し、SI や介護会社（BE）へ通知
9. SI 上のタスク 3（宅内情報の解析）は HGW から受信した宅内情報を解析し、異常があった場合は介護会社（BE）や病院（SI）、外部のサービス利用者へ通知

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 家電機器や住宅設備などの宅内機器からデータを収集

タスク 2 宅内で収集したデータの圧縮

タスク 3 サービス利用者の異常を検知して外部へ通知

タスク 4 外部から宅内機器の制御

家電などの故障対応コールセンター

家電などの故障対応コールセンターのサービスは、住宅機器を遠隔から監視することにより故障対応を行うことが可能である。監視を行う共同のコールセンターは宅内に設置された機器の情報を取得することにより、機器の状態を把握することが可能である。このサービスの概要を図 6.12 に示す。

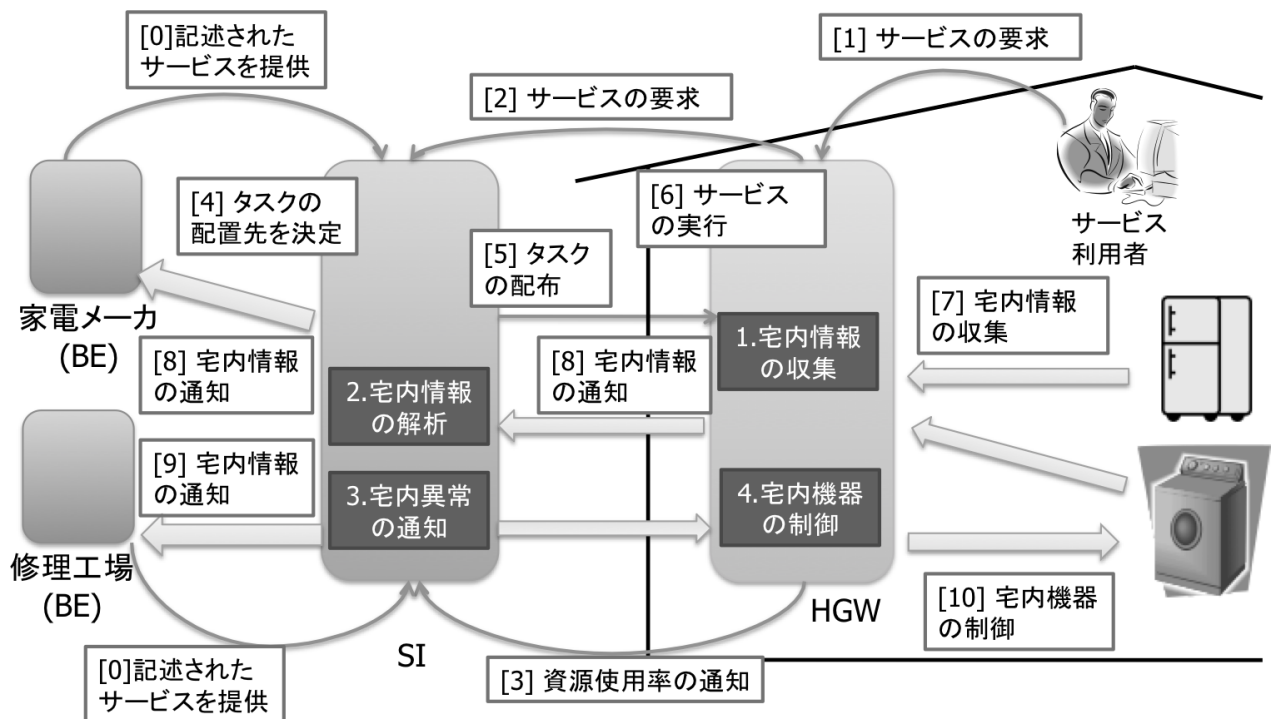


図 6.12: 家電などの故障対応コールセンター

SI モデルを利用した場合、サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1 (宅内情報の収集) とタスク 4 (宅内機器の制御) を配布し、SI にタスク 2 (宅内情報の解析) とタスク 3 (宅内異常の通知) を配置
6. タスクが実行されサービスが開始

7. HGW 上のタスク 1（宅内情報の収集）は宅内機器（家電機器や住宅情報など）からデータを収集
8. HGW から SI へ収集したデータを送信し，SI 上のタスク 2（宅内情報の解析）は受信したデータを解析し得られた宅内情報を管理
9. SI 上のタスク 3（宅内異常の通知）は解析した宅内情報から異常状態を検知すると，家電メーカー（BE）や修理工場（SI），サービス利用者へ通知

また，このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 家電機器や住宅設備などの宅内機器からデータを収集

タスク 2 住宅設備に関する情報の抽出

タスク 3 サービス利用者の異常を検知して外部へ通知

タスク 4 外部から宅内機器の制御

緊急地震速報との連携による地震初期対応の自動化

緊急地震速報との連携による地震初期対応の自動化のサービスは，緊急地震速報システムとの連携により二次災害の抑止や災害後の救助のためのサービスを実現する。緊急地震速報を受信すると，ドアを開放したり，熱源の機器をオフにするなどの機器同士が連携して動作するサービスを提供する。また，宅内に設置されたセンサや家電機器の情報を基に宅内に人がいるかを確認し，外部へ通知することも実現可能である。このサービスの概要を図 6.13 に示す。

SI モデルを利用した場合，サービスは以下の手順で実行される。

1. サービス利用者は HGW を利用してサービスを要求
2. HGW は SI へサービスのリクエストを送信
3. SI は HGW から資源使用率を取得
4. SI はサービスを実現するためのタスクの配置先を決定
5. SI は HGW へタスク 1（宅内機器の制御）とタスク 4（宅内情報の収集）を配布し，SI にタスク 2（宅内情報の通知）を配置
6. タスクが実行されサービスが開始
7. 気象庁（BE）から送信された緊急地震速報を SI や HGW が受信

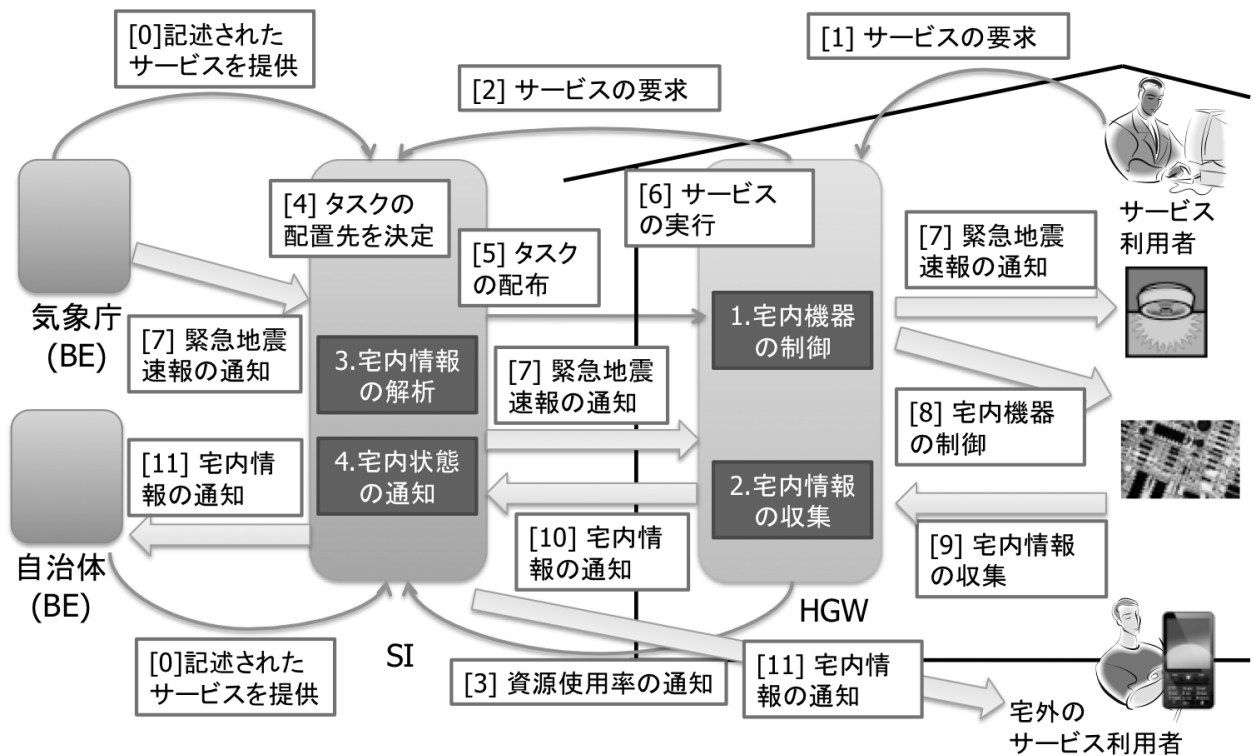


図 6.13: 緊急地震速報との連携による地震初期対応の自動化

8. HGW 上のタスク 1（宅内機器の制御）で家電機器や住宅設備など宅内機器を自動的に制御
9. HGW 上のタスク 2（宅内情報の収集）で宅内機器から情報を収集し、SI へ通知
10. HGW から SI へ収集した宅内情報を通知し、SI 上のタスク 3（宅内情報の解析）でサービス利用者の状態を解析
11. SI から自治体（BE）や外部のサービス利用者へ抽出した情報を通知

また、このサービスを実現するために以下のタスクが実行される必要がある。

タスク 1 外部から宅内機器の制御

タスク 2 家電機器や住宅設備などの宅内機器からデータを収集

タスク 3 住宅設備に関する情報の抽出

タスク 4 サービス利用者の状態を検知して外部へ通知

6.3.2 一般的サービスで実行される共通のタスク

6.3 節でまとめたサービスの例では，ホームネットワークを利用したサービスとして今後考えられている領域について説明した．これらのサービスにおいて本研究で提案するタスクの配置先決定手法が適用可能であれば，一般的なホームネットワークを利用したサービスでは適用可能であることが説明できる．

共通するタスクを分別すると，以下の4つのカテゴリに分けることが可能である．

- 情報収集カテゴリ
- 情報処理カテゴリ
- 情報通信カテゴリ
- 機器制御カテゴリ

情報収集カテゴリ

情報収集カテゴリは，家電機器の設定情報やセンサから取得された値，カメラの映像などの宅内で得られる情報を収集するタスクである．以下に情報収集カテゴリのタスクを示す．

- 宅内のセンサからセンシングしたデータを取得
- 宅内のカメラから映像を取得
- 宅内機器から状態データを取得
- 健康機器から測定結果のデータを取得
- 宅内機器から操作履歴を取得
- 宅内機器の消費電力を取得

情報処理カテゴリ

情報処理カテゴリは，宅内で収集した情報や BE などの外部からの情報に対して計算などの加工を行うことである．以下に情報処理カテゴリのタスクを示す．

- カメラ映像の圧縮，変換
- 外部（BE(警備会社)，携帯電話など）から送信される映像の圧縮，変換
- センサからユーザ情報の抽出，異常検知
- 宅内機器の操作履歴からサービス利用者の嗜好・行動パターンを抽出

情報通信カテゴリ

情報通信カテゴリは、宅内で収集した情報や BE などの外部からの情報を送受信するタスクである。データによって様々なタスクが存在し、以下に情報通信カテゴリのタスクを示す。

- 外部 (BE(警備会社), 携帯電話など) へ宅内情報を送信
- 外部とのコミュニケーション共有 (音声やメッセージなど)
- コンテンツのダウンロード, アップロード
- 外部 (携帯電話や PC, PDA など) 及び宅内機器へコンテンツのストリーミング配信
- 外部からの映像を受信
- 外部へ宅内情報を送信
- 外部から宅内機器の制御命令を受信

機器制御カテゴリ

機器制御は、宅内に設置された家電機器や住宅設備に対して制御メッセージを送信するタスクである。以下に機器制御カテゴリのタスクを示す。

- 宅内の警報装置を制御
- 家電機器の制御 (電源の ON/OFF や設定変更など)

6.3.3 一般的なサービスへの適用

サービスのカテゴリとの関係

一般的なサービスへの適用のため、前節でサービスの例を基にサービスを実現するために実行されるであろうタスクについて調査した。その結果、タスクはその機能によって4つのカテゴリに大別することが可能であることが分かった。カテゴリ分け可能な機能として、情報収集カテゴリ、情報処理カテゴリ、情報通信カテゴリ、機器制御カテゴリがある。この機能によって、どの資源をどれだけ使用するかは大きく異なる。これらの特徴は実際のタスクによって異なるがカテゴリ毎の特徴における大きな違いとして以下のように表すことが可能である。

情報収集カテゴリ r_{dsk}, r_{dskio} への負荷が大きい

情報処理カテゴリ cpu, r_{mem}, r_{diskio} の負荷が大きい

情報通信カテゴリ $dsk, r_{diskio}, r_{iface}$ の負荷が大きい

機器制御カテゴリ どの資源も負荷が小さい

情報収集カテゴリは宅内や宅外に設置された機器から情報を取得する機能を持つため、 r_{dsk}, r_{diskio} への負荷が大きいと考えた。情報処理カテゴリは宅内で収集した情報や BE などの外部からの情報に対して圧縮などの加工を行う機能を持つため、 cpu, r_{mem}, r_{diskio} への負荷が大きいと考えた。情報通信カテゴリは宅内で収集した情報や外部からの情報を送受信する機能を持つため、 $dsk, r_{diskio}, r_{iface}$ への負荷が大きいと考えた。機器制御タスクは宅内に設置された家電機器や住宅設備に対して制御命令を行う機能を持ち、資源に対する負荷はどれも小さいと考えた。

このためサービスを実現する各タスクは、カテゴリの特性を考慮して特徴係数を設定することが可能であると考えられる。そのため、特徴係数を利用してタスクコストを求めることが可能となり、そのことから適性スコアを利用した提案手法の有効性が高くなることが考えられる。

ホームネットワークを利用したサービスの形態との関係

また、提案手法ではサービス（タスク）が繰り返し実行されることにより要素の性能や状態などの特徴が学習される。過去に学習した情報を基に、特徴係数やタスクコストなどのパラメータは適切な値に収束する。このため、ホームネットワークのサービスのような長期的に運用されるシステムに対して、提案手法は有効に作用することが期待される。

6.4 考察のまとめ

本章では、提案したタスクコストや特徴係数、適性スコアの有効性について実験結果を基に評価を行った。また、一般的なホームネットワークを利用したサービスについて調査し適用性について検証した。

まず、タスクコストと特徴係数の有効性を評価するために、タスクを一つ実行した場合と複数実行した場合の二通りの場合の実験を行った。結果を比較すると、複数のタスクが実行されている場合には、特徴係数の初期値に適切な値を設定することがより重要であることが判明した。しかし、実験結果からタスクが繰り返し実行されることにより特徴係数は適切な値に収束していることが判明した。このため、繰り返し実行することで正確なタスクコストを求めることが可能であることが判明した。すなわち、ホームネットワークを利用したサービスのように長期的に運用されるシステムに対して、提案手法は有効に作用すると言える。

また、適性スコアの有効性を評価するため、実験時の実行時間を計測した。結果を比較すると、適性スコアで選択した要素の方が実行時間が長くなる場合があることが判明した。これは、複数のタスクを実行するときに、同時に実行したタスクが実行時間に影響を与えたためと推測される。このため、提案手法では考慮している資源に加えてさらに実行時間についても考慮することで、より有効性の高い手法となると考えられる。

最後に、一般的なサービスへの適用性を検証するため、現在議論されているホームネットワークを利用したサービスを調査した。調査した結果、タスクは機能毎にカテゴリ分けすることが可能であることが判明された。そして機能毎にどの資源をどれだけ使用するかをある程度決定することが可能であると考えられる。すなわち、カテゴリ毎に特徴係数を設定することが可能であるといえ、提案手法は一般的なサービスへ適用する可能性は高いと考えられる。

第7章 まとめと今後の課題

現在、ホームネットワークを利用した様々なサービスが実現されつつある。サービスの実現には、サービス提供者、中間サービス事業者、ホームゲートウェイの3つの要素で構成されたシステムを利用している。これらの要素で構成されたシステムを本研究ではSIモデルと定義した。SIモデル上の各要素で単数あるいは複数のタスクを実行することによりサービスを実現可能であるが、要素の持つ資源は要素毎に異なるためタスクの実行により資源の不足が発生する可能性がある。このため本研究では、要素の持つ資源と実行されるタスクの性質を基にして、タスクの配置先決定手法を提案した。提案手法を評価するため、実際のホームネットワークを利用したサービスを模倣したシミュレーションによる評価実験を行った。また、現在考案されている様々なホームネットワークを利用したサービスを調査し、提案手法が一般的なサービスへ適用可能であるか検証した。

本論文では、SIモデルが登場する背景について説明し、SIモデルの必要性について説明した。そして、SIモデル上の要素で必要となる内部構成を提案し、ホームネットワークを利用したサービスがSIモデル上でどのように実現されるかについて説明した。また、タスク配置手法としてオペレーティングシステムで利用されている一般的な手法を調査し、本研究との関連性について調査を行った。また、負荷に対する問題を解決するために負荷分散に関する関連研究の調査を行い、負荷分散装置で利用されている負荷分散技術について調査した。負荷分散装置で利用されている手法を参考とし、資源を考慮して算出される適性スコアを利用したタスク配置手法を提案した。監視する資源について定義し、適性スコアを導出するために必要となるパラメータについて定義した。

提案した適性スコアによるタスク配置手法の有効性を評価するため、PCを利用したシミュレーションによる評価実験を行った。実験結果より、適性スコアはタスクの配置先としての適性を定量的に示すことが可能であることが判明した。適性スコアを利用することで負荷を考慮したタスクの配置先を決定することが可能であり、提案手法の有効性を確認できた。また、サービスの実行時間を比較した場合では、適性スコアによって選択された要素の方が選択しない要素の実行時間より大きくなる場合がある。これは提案手法において資源の負荷については考慮していたが、実行時間については考慮していないことが原因である。よって提案手法では、資源に加えてサービスの実行時間についても考慮することで、より有効性の高い手法となると考えられる。

本研究で提案するタスクの配置先決定手法を利用することで、SIモデル上の構成要素において資源不足が発生する可能性を軽減し資源を有効に活用することが可能になる。

今後の課題として、タスクの所要時間を考慮した場合やタスクのプリエンプションや動

的なタスクの移送を考慮した場合、タスクのデッドラインを考慮した場合の手法を考案することが挙げられる。また、実サービスの動作する環境で提案手法を実験することで実世界での有効性を検証することも挙げられる。また提案手法によって資源を考慮してタスクを配置した場合と、資源を考慮せずにタスクを配置した場合の比較を行い、ホームネットワークを利用したサービスの実行に影響が出るか調査することが挙げられる。

謝辞

本論文の執筆につぎまして、研究に対する御指導を賜りました丹康雄教授に心から深く感謝すると共に、ここに御礼申し上げます。適切なご助言を頂きました Lim Azman Osman 准教授に深く感謝致します。

また、貴重なご助言やご意見を頂いた中田潤也氏、岡田崇氏、金準修氏、Marios Sioutis 氏には深く感謝の言葉を申し上げます。

最後に、学業や学生生活において苦楽を共にした友人達と、学生生活を最後まで支えてくれた両親に深く感謝致します。

参考文献

- [1] A.S. タネンバウム (著), 水野忠則 (翻訳). 分散オペレーティングシステム. ピアソン・エデュケーション, 2003.
- [2] A.S. タネンバウム (著), 千輝順子 (翻訳). オペレーティングシステム. プレンティスホール, 1998.
- [3] F5 Networks Inc. Overview of Dynamic Ratio load balancing. <https://support.f5.com/kb/en-us/solutions/public/9000/100/sol9125.html>, 20100129.
- [4] ITU-T Recommendation J.190. Architecture of MediaHomeNet that supports cable-based services, 2002.
- [5] OSGi Alliance. OSGi Alliance. <http://www.osgi.org/>, 20100201.
- [6] Thomas Decker. Dynamic load balancing and scheduling. <http://www2.cs.uni-paderborn.de/cs/ag-monien/RESEARCH/LOADBAL/>, 20100129.
- [7] 塩谷康夫, 太田学, 片山薫, 石川博. P2P による転送遅延を考慮した静的負荷分散方式. 信学技報. 社団法人電子情報通信学会, 20050301.
- [8] 今井智大. ホームネットワークサービスアーキテクチャにおける障害検出・回復モデルに関する研究. Master's thesis, 北陸先端科学技術大学院大学, 2009.
- [9] 今井智大, 岡田崇, 中田潤也, 丹康雄. ホームネットワークにおける障害検知・回復を考慮したサービス提供モデル (セッション 1: 慶應義塾大学). 情報処理学会研究報告. UBI, [ユビキタスコンピューティングシステム], Vol. 2009, No. 17, pp. 1–8, 20090225.
- [10] 西村健治, 上野仁, 山本幹, 池田博昌. ネットワーク遅延を考慮した動的負荷分散方式. 電子情報通信学会論文誌. B, 通信, Vol. 82, No. 10, pp. 1763–1772, 19991025.
- [11] 沢野泰淳, 張勇兵, 高木英明. 移動エージェントを用いたネットワーク負荷分散システムの構築とその評価. 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, Vol. 84, No. 9, pp. 1450–1453, 20010901.

- [12] 丹康雄監修, 宅内情報通信放送高度化フォーラム編. ホームネットワークと情報家電. オーム社, 2004.
- [13] 藤倉俊幸. リアルタイム/マルチタスクシステムの徹底研究. CQ 出版社, 2004.
- [14] 次世代 IP ネットワーク推進フォーラム ホームネットワーク WG 普及促進 SWG. 次世代ホームネットワークが描く新たな価値進化時代へ向けた挑戦. http://ngnforum.nict.go.jp/kenhyoubukai/homenetworkwg/homenetworkwg27th/HNWG2008-HNWG.P_v1_rev2.pdf, 20100129.
- [15] 野口健一郎. オペレーティングシステム. オーム社, 2004.
- [16] 鈴木伸夫, 山本幹, 岡田博美, 池田博昌. 分散システムにおけるプロセスバッファを用いた動的負荷分散方式. 情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム], Vol. 95, No. 59, pp. 97–102, 19950622.