

サーバのログイン時間とユーザの歩数を使用した作業時間の予測による物理マシンの停止と消費電力の削減

坂井 萌桜¹ 山野 倅平¹ 串田 高幸¹

概要: 東京工科大学コンピュータサイエンス学部の Cloud and Distributed Systems Laboratory には、物理マシンが 10 台ある。そのうち、CDSL に所属している学生が使用できる物理マシンは 7 台あり、それらは学生が常に仮想マシンで作業を行えるよう 24 時間稼働している。課題は、ユーザが物理マシン内の仮想マシン上の OS へ SSH 接続していない時間帯にも物理マシンが起動しており、電力を消費していることである。提案では、ユーザの歩数データと物理マシン内の仮想マシン上の OS への SSH 接続の接続時間のデータをもちいて、ユーザがアクセスする可能性を示す値とユーザがアクセスしない可能性を示す値を設定し、2 つの値を比較することで物理マシンが使用されない時間の予測を行い、物理マシンの停止スケジュールを作成し、物理マシンの停止、起動を行うことで消費電力を削減する手法を提案する。評価実験は提案したアルゴリズムをもちいて作成した物理マシンの停止スケジュールの正解率について 3 つの項目を評価する。3 つの項目とは、物理マシンが使用されていない時間に停止可能な時間、削減可能な電力量、停止と予測した時間にユーザが作業した時間である。評価は作成した停止スケジュールとユーザからのアクセスを比較し、スケジュールの正解率を計算することで行った。その結果、物理マシンが未使用の時間に物理マシンを最大で約 65.66%、最小で約 49.19% 停止できた。停止すると予測した時間のうち、約 0.90% の時間には仮想マシンが使用されていた。電力では各物理マシンの平均で電力消費の約 56.21% の電力の削減をすることができた。

1. はじめに

背景

世界の電力消費量は年々増加している。その理由の 1 つに、クラウドサービスの普及がある [1]。クラウドサービスの例として、Google Cloud や Amazon Web Service があげられる。データセンターとは、サーバやネットワーク機器を設置、運用するための施設であり、その運用のために多大な電力を消費している [2]。国際エネルギー機関の調査によると、世界中のデータセンターの電力消費量は全世界の電力消費の約 1% に及んでいる [3]。データセンターは IT 業界を支える基盤であり、需要が高まっているため、その電力消費量は将来的に増えていくことが想定される [4]。

東京工科大学コンピュータサイエンス学部の Cloud and Distributed Systems Laboratory (以下 CDSL と略す) には、物理マシン (以下 PM と略す) が 10 台ある。これらの PM 内では、VMware 社が提供している ESXi が配置されている。2 台は基幹サーバ、1 台は外部公開用のサイトを運用するために使用されている。そのため、CDSL に所属し

ている学生が使用できる PM は、Jasmine, Rose, Plum, Violet, Mint, Lotus, Lily の 7 台ある。

Virtual Machine (以下 VM と略す) は、PM 上で動作するハイパーバイザーによって PM の CPU、メモリが仮想化され、それぞれが独立したコンピューティング環境として機能する。学生は VM を自由に PM 内の ESXi 上に構築、運用している。いつでも VM 上の OS へ SSH 接続することができるように、PM は常に起動している。

課題

課題は、ユーザが PM 内の VM 上の OS へ SSH 接続していない時間帯にも PM が起動しており、電力を消費していることである。

ユーザによる PM 内の VM 上の OS への SSH 接続と PM の電力消費について、図 1 に示す。ユーザが VM 上の OS へ SSH 接続していない場合でも、PM はユーザが VM 上の OS へ SSH 接続している場合と同様に起動しており、電力を消費している。

¹ 東京工科大学コンピュータサイエンス学部
〒192-0982 東京都八王子市片倉町 1404-1

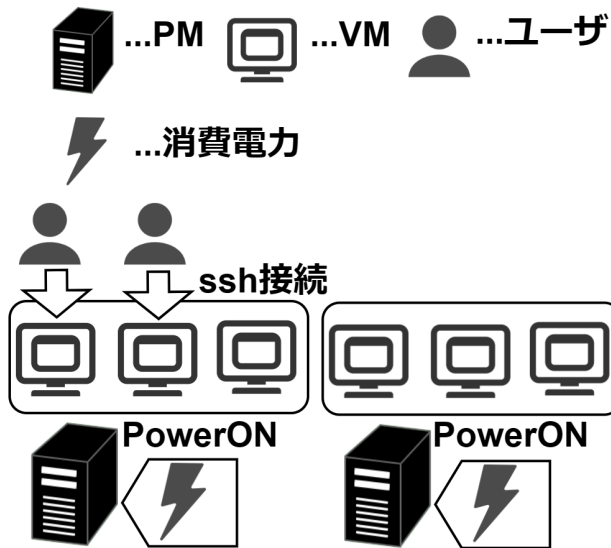


図 1 ユーザによる PM 内の VM の使用と PM の消費電力

各章の概要

第 2 章では関連研究について述べる. 第 3 章では提案方式とユースケースシナリオについて述べる. 第 4 章では実装について述べる. 第 5 章では実験環境や, 提案方式と実験結果についての評価を述べる. 第 6 章では議論を行う. 第 7 章は本稿におけるまとめである.

2. 関連研究

VM を効率的に配置することで PM の過負荷, 低負荷の可能性を減らし, PM の電力消費を削減する論文がある. この論文では, 複数の PM を比較し, エネルギー消費が最小になる PM を選択して VM を配置する ERAP アルゴリズムを提案している. この手法では, 動的に VM を別の PM に移行することができる環境における消費電力の削減が課題となっている [5].

複数台の PM を, 個々の PM ではなく PM の集合体と捉え, 集合体全体の電力管理を一括で行うアンサンブル型電力管理を行うことで, PM の消費電力を削減する論文がある. この論文では, PM 全体の消費電力を監視, 管理するコントローラーがリアルタイムで個々の PM の消費電力データを収集し, アンサンブル全体での消費電力を計算, その後ポリシー駆動型の制御エンジンを使用し, 電力管理の指示を各 PM に送信する提案を行っている. この手法は, 電力使用のピーク時における消費電力に注目し, 消費電力の削減を行っている. ピーク時だけでなく, 平常時の消費電力にも注目したより総合的な管理手法を検討していないという点において改善の余地がある [6].

データセンターの消費電力を削減するために, 直接液体冷却技術をもちいて効率的な熱管理を行う論文がある. この論文ではデータセンターの消費電力の削減を熱管理の観点から試みている. 空冷ヒートシンクや冷却プレート为例

とした従来の冷却技術では, データセンターの性能維持のための熱管理に電力がかかり, データセンター全体の消費電力が増加する. シリコン表面の背面に直接冷却液を接触させる技術である, 直接冷却技術をもちいることで, 従来の冷却方法と比較し, 消費電力を削減している. CDSL で使用されているような小規模の PM はデータセンターと比べて全体的な消費電力が低く, 導入費用に対する消費電力の削減が見込めない [7].

データセンターにおける大規模ストレージシステムにおいて, 使用されていないディスクをスリープ状態にすることで消費電力を削減する論文がある. この論文では, アクセス頻度の低いデータをスリープ状態にする MAID アルゴリズムと, データのアクセス頻度にもとづき, アクセス頻度の高いデータを特定のディスクに集中させる PDC アルゴリズムを提案している. この手法は, ストレージ全体の消費電力を大幅に削減可能だが, ディスクの起動とスリープの切り替えに伴う遅延や, PDC アルゴリズムにおけるデータ移行に伴うオーバーヘッドが生じる点において改善の余地がある [8,9].

3. 提案

本稿の提案では, ユーザの歩数データと PM ごとの VM 上の OS へのアクセス時間のデータをもちいて PM が使用されていない時間を予測し, PM の停止スケジュールを作成することで PM の消費電力を削減する手法を提案する. VM へのアクセス時間とは, ユーザが VM 上の OS への SSH 接続を行っている時間のことを示す. ユーザによる PM 内の VM へのアクセスがない時間を予測する値として, ユーザのアクセスする可能性を示す ProByAccess と, ユーザのアクセスしない可能性を示す ProByStep を設定する. ProByAccess は, PM 内の VM 上の OS へユーザのアクセスがある可能性を示す値であり, 過去の PM 内の VM 上の OS へのアクセス時間のデータをもとに設定する. ProByStep は, PM 内の VM 上の OS へユーザのアクセスがない可能性を示す値であり, 過去の歩数のデータをもとに設定する. 過去 10 週間分のデータから ProByAccess と ProByStep をユーザの曜日ごとに対し 1 分ごとに計算する. 計算した ProByAccess と ProByStep から, ユーザが PM 内の VM 上の OS へアクセスしない時間を予測するスケジュールを作成する. PM の停止と起動は作成したスケジュールをもとに行う.

対象の時刻における歩行の特徴の取得に最適な長さの時間として, ProByAccess と ProByStep の設定を仮に 1 分間隔とした.

ProByAccess と ProByStep を求めるために過去 10 週間分のデータを使用する理由は, 人間の習慣が形成されるまでには平均 66 日かかるからである [10]. 人間の習慣の形成にかかる日数を超える最も少ない週数として, 使用する

データの対象期間を仮に 10 週間とした。

曜日ごとに ProByAccess と ProByStep を求める理由は、曜日ごとに異なる固定のスケジュールが存在するからである。具体的には、アルバイトや研究室のコアタイム、大学の講義があげられる。これらのスケジュールは、ユーザの VM 上の OS へのアクセス時間に影響を及ぼす。固定のスケジュールによる VM 上の OS へのアクセス時間の変化に対応するため、曜日ごとに分けて値を設定することとした。提案方式の概要を図 2 に示す。

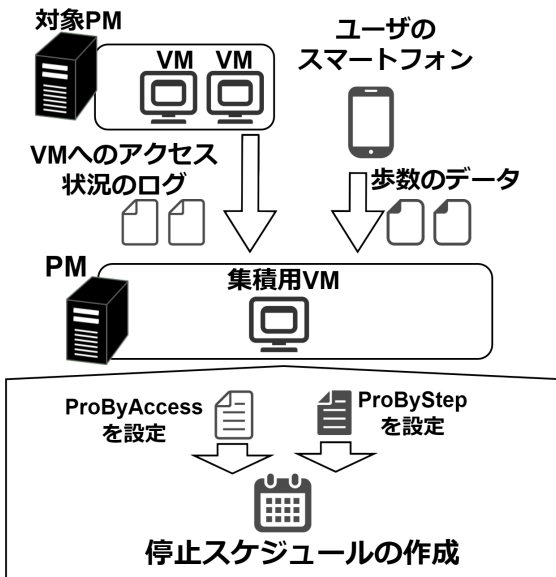


図 2 提案の概要

PM ごとの VM 上の OS へのアクセス状況のログを集積用サーバに集積する。歩数のデータをユーザのスマートフォンから取得し、集積用サーバに収集する。集積したアクセス状況のログから ProByAccess を設定する。収集した歩数のデータから ProByStep を設定する。1 分ごとに ProByAccess と ProByStep の比較を行い、ProByAccess が ProByStep 以上ならアクセスがある、ProByAccess が ProByStep 未満ならアクセスはないと予測し、PM の停止スケジュールを作成する。PM の停止と起動は停止スケジュールをもとに行う。図 2 の作業を PM ごとに行う。

VM 上の OS へのアクセス状況のログの集積

CDSL 内で稼働している 10 台の PM のうち、ユーザが自由に VM を作成、運用することができる 7 台の PM である、Jasmine, Rose, Plum, Violet, Mint, Lotus, Lily を対象とする。PM ごとに作成されている全ての VM で、10 分間隔で last コマンドと who -u コマンドを実行し、その実行結果を集積用サーバに送る。コマンドの実行時間、重複のないアクセスユーザ、アクセス人数をログ集積ツールを用いて、集積用サーバへと送信し、VM ごとに記録する。

歩数の収集

CDSL 内のユーザのスマートフォンに Google Fit をインストールし、そこで記録された歩数データを対象とする。Google Fit とは、Google が提供している健康管理アプリである。主にフィットネス活動や健康状態のモニタリングをサポートするために提供されている。

PM 内に作成されている VM を使用しているユーザの歩数データを収集する。Google Fit をもちいて集積した歩数データは GoogleTakeout を利用し CSV ファイルとして取得する。取得した CSV ファイルをログを集積している VM に送信する。

VM 上の OS へのアクセス状況のログをユーザごとに整理

集積しているアクセス状況のログから、PM ごとの全ての VM 上の OS へのアクセス状況のログを直近の 10 週間分取得する。取得した VM 上の OS へのアクセス状況のログをユーザごとに分けた後、曜日ごとに分ける。PM 内に複数の VM を持つユーザは、そのユーザの保持している全ての VM のいずれかにアクセスがあった場合にアクセスがあったものとして扱い、10 週間分のアクセス状態の回数を 1 分ごとに、10 週間分合計する。

歩行状況をユーザごとに整理

Google Fit で集積している歩数のデータから、PM 内の VM を使用しているユーザの歩数のデータを直近 10 週間分取得する。取得したユーザの歩数のデータを曜日ごとに分け、10 週間分の歩数のデータから対象ユーザの曜日ごとの 1 分間の平均歩数を求める。その曜日において 1 分間の平均歩数以上の歩数がある 1 分を歩行していた 1 分として扱い、歩行していた回数を 1 分ごとに 10 週間分合計する。

アクセスする可能性を示す値の設定

PM ごとの VM 上の OS へユーザがアクセスする可能性を示す値である ProByAccess と ProByStep を設定する。ProByAccess と ProByStep は曜日ごと、ユーザごと、1 分ごとに設定する。

ProByAccess は過去 10 週間分の PM 内の VM 上の OS へのユーザのアクセスをもとに設定する値である。過去に VM 上の OS へアクセスがあった時刻は、今後もアクセスする可能性がある時刻と判断する。

ProByStep は過去 10 週間分の歩数のデータをもとに設定する値である。対象ユーザの過去の平均的な歩数よりも歩数の多い時間帯を歩行がある時間帯とする。歩行がある時間帯は、移動時間であり、今後も VM 上の OS へアクセスする可能性がない時間帯と判断する。ユーザのアクセス可能性を示す 2 つの値を比較し、対象の時刻に PM 内の VM 上の OS へユーザのアクセスがあるかを予測する。

ProByAccess の設定

ProByAccess を 1 分ごとに設定する。ProByAccess は PM 内の VM 上の OS ヘューザのアクセスがある可能性を示す値であり、PM 内の VM 上の OS へのアクセス状況のログをもとにユーザごとに求める。

時刻 t に対して ProByAccess を求めたものを、ProByAccess(t) と設定する。まず、時刻 t における過去 10 週間分の PM 内の VM 上の OS へのアクセス状態の回数を合計する。アクセス状態とは、ユーザが VM 上の OS に SSH 接続し、そのセッションが継続している時間を示す。時刻 t にアクセス状態の場合、時刻 t のアクセス状態の回数は 1 回とする。アクセス状態の回数は曜日ごとに合計するため、時刻 t における過去 10 週間分のアクセス状態の回数の最大値は 10 回である。

次に、アクセス状態の回数の割合を求める。時刻 t における過去 10 週間分のアクセス状態の回数が 3 回の場合、時刻 t におけるアクセス状態の回数の割合は 30% である。時刻 t におけるアクセス状態の回数の割合を ProByAccess(t) として、1 分ごとに ProByAccess(t) を求める。

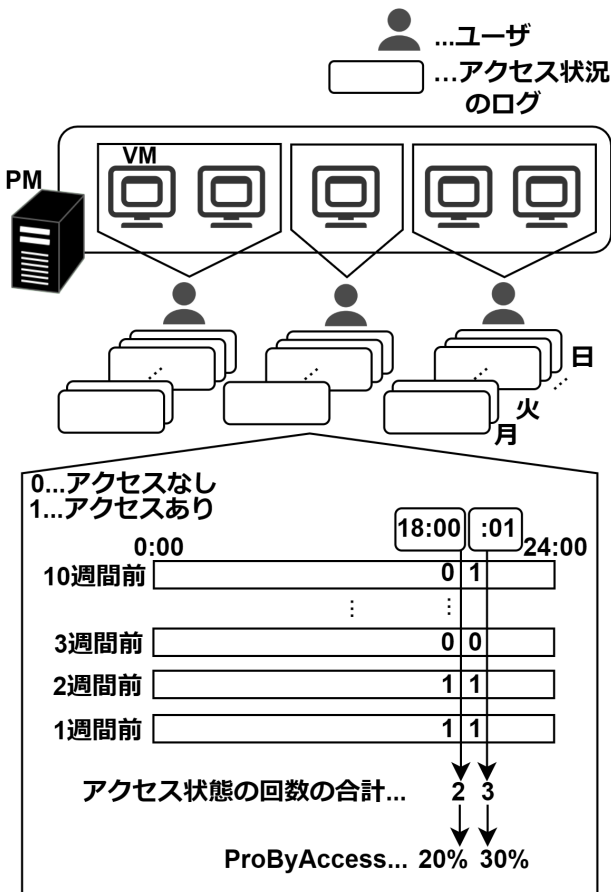


図 3 ProByAccess の設定の流れの例

図 3 は ProByAccess を設定する流れの一例である。

例えば、ユーザの火曜日の 18:00 における ProByAccess を求める場合、ProByAccess(t) は ProByAccess(18:00) と

表す。アクセスありの状態を 1 回、アクセスなしの状態を 0 回としてアクセス状態の回数を足し合わせる。過去 10 週間分の火曜日の 18:00 におけるアクセス状態の回数の合計が 2 回だった場合、ProByAccess(18:00) は 20% である。1 分後の 18:01 においては、ProByAccess(t) は ProByAccess(18:01) であり、その時刻における過去 10 週間分のアクセス状態の回数の合計が 3 回だった場合、ProByAccess(18:01) は 30% となる。ユーザの水曜日の 18:00 における ProByAccess を求める場合は、過去 10 週間分の水曜日の 18:00 におけるアクセス状態の回数の合計から ProByAccess(18:00) を求める。

PM 内に VM を所持している全てのユーザに対して、ProByAccess を求める。ユーザが PM 内に複数の VM を持つ場合は、1 人のユーザとして 1 分ごとに ProByAccess(t) を設定する。

ProByStep の設定

ProByStep を 1 分ごとに設定する。ProByStep は、歩数のデータをもとに設定する値であり、PM 内の VM 上の OS ヘューザのアクセスがない可能性を示す。この値はユーザごとに求める。

時刻 t に対して ProByStep を求め、ProByStep(t) と設定する。ProByStep(t) は以下のように求める。まず、ユーザごとに曜日ごとの 1 分間における平均歩数を求める。過去 10 週間分の歩数のデータを曜日ごとに分けて平均歩数を求めるため、1 ユーザに対して求める平均歩数は 7 個である。次に、特定の曜日における時刻 t の歩数と、その曜日の平均歩数を比較する。時刻 t における歩数の方が平均歩数よりも多い場合、時刻 t においてユーザは移動していたと判断する。そして、時刻 t における過去 10 週間分の移動していた回数を合計する。曜日ごとに分けて合計するため、時刻 t における過去 10 週間分の移動していた回数の最大値は 10 回である。最後に、移動していた回数の割合を求める。1 分ごとに、時刻 t における移動していた回数の割合を示す ProByStep(t) を求める。ProByStep の設定の流れの一例を図 4 に示す。

例えば、ユーザの火曜日の 18:00 における ProByStep を求める場合、ProByStep(t) は ProByStep(18:00) と表す。まず、火曜日における 1 分間の平均歩数を求める。平均歩数が 50 歩の場合、50 歩以上歩いた 1 分は移動があると判断し、50 歩以上歩いていなければ移動がないと判断する。過去 10 週間における歩行していた回数を足し合わせる。過去 10 週間分の火曜日の 18:00 における歩行していた回数の合計が 1 回だった場合、ProByStep(18:00) は 10% である。1 分後の 18:01 においては、ProByStep(t) は ProByStep(18:01) であり、その時刻における過去 10 週間分の歩行していた回数の合計が 4 回だった場合、ProByStep(18:01) は 40% となる。ユーザの水曜日の 18:00 における ProByStep を

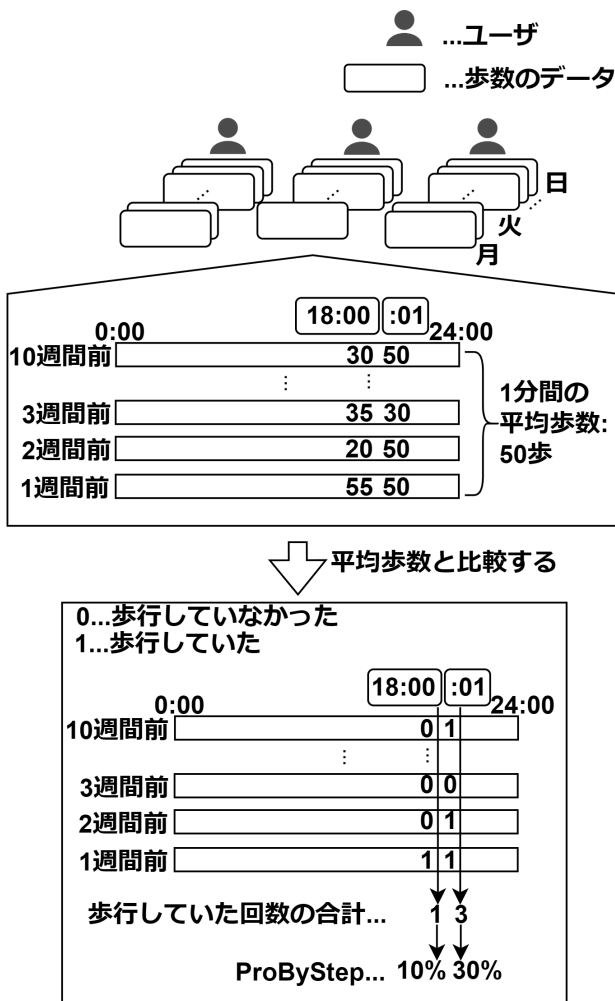


図 4 ProByStep の設定の流れの例

求める場合は、過去 10 週間分の水曜日の 18:00 における歩行していた回数の合計から ProByStep(18:00) を求める。

ユーザごとの PM 停止スケジュールの作成

ProByAccess(t) と ProByStep(t) をもちいて、PM ごとにユーザごとの PM 停止スケジュールを作成する。ユーザごとの PM 停止スケジュールは、ユーザが作業を行わない時間を予測し、PM を停止する事ができる時間を設定したものである。ProByStep(t) が ProByAccess(t) 以上の場合、ユーザのアクセスがある可能性を示す値よりもアクセスがない可能性を示す値の方が高いため、時刻 t におけるアクセスはないとし、PM を停止と判断する。そして、時刻 t を PM 停止スケジュールに追加する。実際に作成する停止スケジュールにおける PM の停止から起動までの時間は 10 分以上とする。判定を 10 分以上の間隔をあけて行う理由は、PM を起動してから停止するまで少なくとも 10 分以上稼働させることが推奨されているからである*1。

*1 <https://infohub.delltechnologies.com/en-us/p/idrac9-virtual-power-cycle-remotely-power-cycle-dell-emc-powered-edge-servers/>(参照 2024/12/23)

このため、ProByAccess(t) と ProByStep(t) をもちいて 1 分間隔で PM の起動と停止を判定し、PM 起動の判定後 10 分以内に停止の判定がある場合においては、停止をしない判定を行い、ユーザごとの PM 停止スケジュールを作成する。

ユーザの ProByStep と ProByStep を比較して、ユーザごとの PM 停止スケジュールを設定する流れの一例を図 5 に示す。

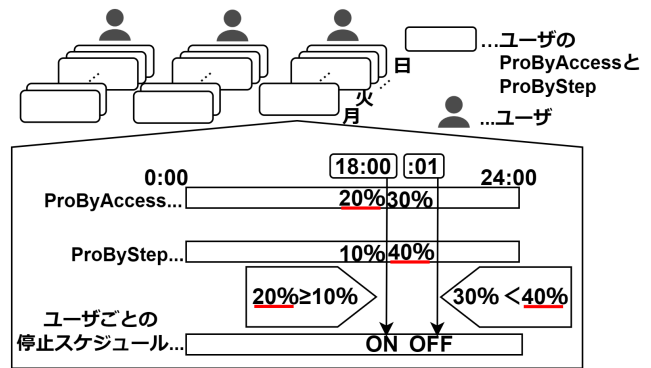


図 5 ユーザごとの PM 停止スケジュールの作成

例えば、ユーザの火曜日の 18:00 における PM の停止スケジュールを求める場合、ProByAccess(18:00) と ProByStep(18:00) を比較する。ProByAccess(18:00) が 20%、ProByStep(18:00) が 10% の場合、ProByStep(18:00) が ProByAccess(18:00) 以上ではないため、2 つの値を比較して、ユーザごとの PM 停止スケジュールを ON 状態と判断する。18:01 においては、ProByAccess(t) と ProByStep(t) はそれぞれ ProByAccess(18:01) と ProByStep(18:01) であり、ProByAccess(18:10) は 30%、ProByStep(18:10) は 40% のため、2 つの値を比較して、ユーザごとの PM 停止スケジュールを OFF 状態と判断する。18:01 において、ユーザごとの PM 停止スケジュールは OFF 状態に設定されているが、18:01 時点でユーザごとの PM 停止スケジュールにおいて PM が 10 分間連続して稼働していない場合は、ON 状態と判断する。

PM ごとの停止スケジュールの作成

ユーザごとに作成した PM 停止スケジュールをもとに、PM の停止スケジュールを作成する。PM の停止スケジュール作成の流れを図 6 に示す。

PM 内の VM を所持している全てのユーザの停止スケジュールを 1 分ごとに参照する。対象の 1 分において、PM 内の VM を所有しているユーザの停止スケジュールがすべて OFF 状態の場合のみ、PM 全体の停止スケジュールに追加する。ユーザの停止スケジュールが、1 人以上 ON 状態の場合、PM 内の VM 上の OS へのアクセス可能性がある 1 分であると判断し、PM 全体の停止スケジュールには

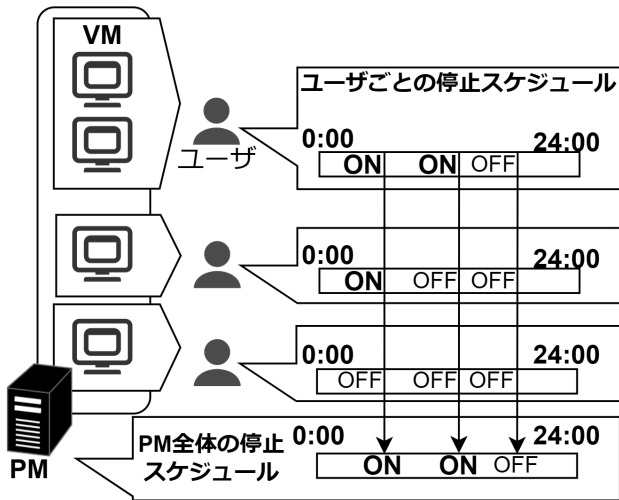


図 6 PM ごとの停止スケジュールの作成

追加しない。PMの停止、起動は、作成したPM全体の停止スケジュールに沿って行う。

ユースケース・シナリオ

本稿では、ユースケースとして、CDSLのPM運用環境を想定する。CDSLでは10台のPMが運用されている。そのうちの7台では、CDSLに所属している学生がVMを自由にESXi上に構築、運用している。常時PM内のVM上のOSにSSH接続することができるように、各PMは常に起動している。

図7に、ユースケースシナリオを示す。

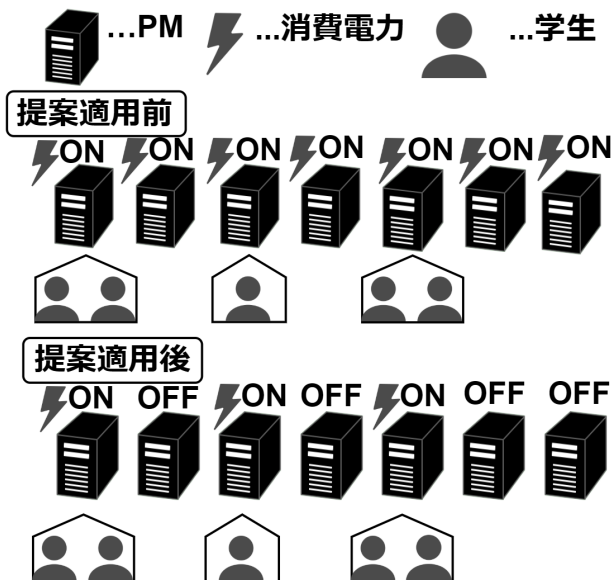


図 7 ユースケースシナリオ

提案の適用前は、各PMは24時間起動しているが、学生がPM内のVM上のOSにSSH接続していない時間がある。この時間はPMで作業が行われておらず、電力を消費している。提案の適用後は、学生がVM上のOSへのSSH

接続を行う時間を予測したPMの停止スケジュールを使用することで、学生がVMを使用していない時間にPMの電源をOFFにすることができる。そのため、提案適用前と比較し、消費電力を削減できる。

4. 実装

提案手法をもとに、Pythonとシェルスクリプトをもちいて提案ツールを実装した。Pythonで使用したライブラリは以下の通りである。

- re
- os
- calender
- xml.etree.ElementTree
- datetime
- collections.defaultdict
- pandas

実装の構成を図8に示す。

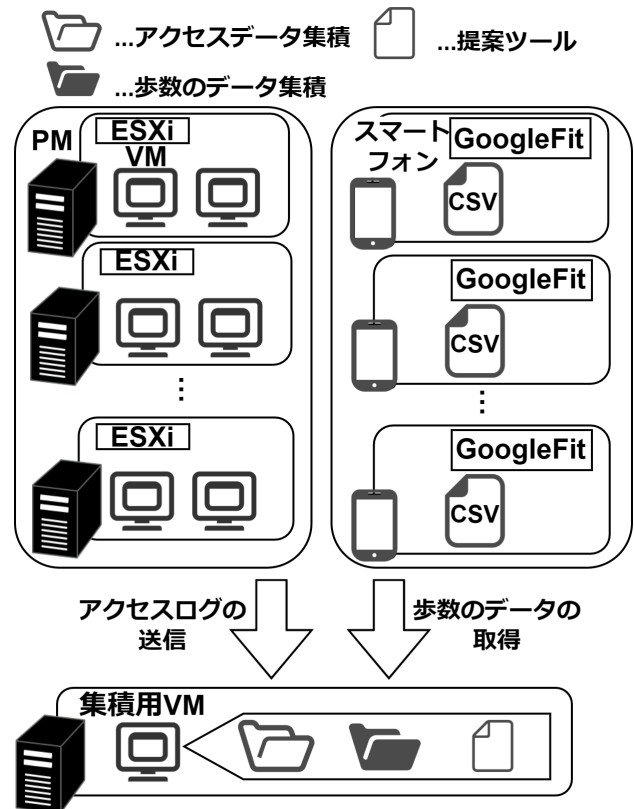


図 8 実装の構成図

提案の対象となるPM内の、VM上のOSで動作するプログラムから、ログを集積するVMへアクセスログを送信する。提案の対象となるPM内のVMを使用しているユーザーの歩数データを取得し、ログを集積するVMで収集する。提案ツールはログを集積するVM内にある。

VM 上の OS へのアクセス状況ログの集積

提案の対象となる PM 内の全ての VM にログ集積ツールを導入する。ログ集積ツールとして、ログを集積する VM へ提案の PM 内に存在する VM からのユーザのアクセスログを送信するシェルスクリプト、log_collect.sh を作成した。

このツールは、導入した VM 内で 10 分間隔で last コマンドと who -u コマンドを実行し、その出力結果をログを集積する VM に送信する。last コマンドでは対象の VM 上の OS へのユーザのアクセス履歴を取得する。who -u コマンドでは、コマンド実行時にアクセスしているユーザとそのアイドル時間を取得する。log_collect.sh は systemd の timer 機能を使用して 10 分間隔で実行する。last コマンドの出力結果を分析し、1 分ごとのアクセス状況を取得することができる。

歩数の収集

提案の対象となる PM 内の VM を使用しているユーザの歩数データを集積する。集積したデータの収集には GoogleTakeout を利用し、日付ごとに分けられた CSV ファイルとして取得した。GoogleTakeout はユーザの利用している Google アカウントに紐づけられたデータをエクスポートする、Google の提供するサービスである*2。ユーザの Google アカウントにログインし、Google Fit で集積されたデータを選択してエクスポートを行い、生成された CSV ファイルをユーザから提供してもらうことで歩数のデータを収集する。収集した CSV ファイルはログを集積する VM へ送信し、ユーザごとに管理する。

VM 上の OS へのアクセス状況ログをユーザごとに整理

ログ集積ツールで集積した VM へのログを、Python をもちいて整理する。PM 内の VM を使用しているユーザごとに分け、その後、曜日ごとに分類する。同一のユーザ、曜日、時間をもつログのアクセス数を合計したデータへ変換する。

曜日ごとの歩数の平均を求める

Python をもちいて、歩数のデータを記録した CSV ファイルから曜日ごとの歩数の平均を求める。Google Fit で収集した歩数のデータは 15 分ごとに記録されているため、15 分間の歩数の平均を求める。

ProByAccess と ProByStep の算出

ユーザごとに分けて整理した PM ごとの VM 上の OS へのアクセス状況のログに ProByAccess を求める提案アルゴリズムを使用する。ProByAccess はユーザごとに求める。PM の停止スケジュール作成の際に比較する歩数の

データが 15 分ごとに記録されているため、提案アルゴリズム内では 1 分ごとに求めていた ProByAccess は 15 分ごとに算出する。ユーザごとに分けて整理した歩数のデータに ProByStep を求める提案アルゴリズムを使用する。ProByStep はユーザごとに求める。歩数のデータは 15 分ごとに記録されており、曜日ごとの歩数の平均も 15 分で求めているため、ProByStep も 15 分ごとに算出する。

ユーザごとの PM 停止スケジュールの作成

15 分ごとに算出されたユーザごとの ProByAccess と ProByStep を比較する。対象の 15 分において ProByStep が ProByAccess 以上の時、停止スケジュールにその 15 分を追加する。曜日ごと、時間帯ごとに比較を行い、ユーザごとの PM 停止スケジュールを PM ごとに設定する。

PM ごとの停止スケジュールの作成

ユーザごとに作成した PM 停止スケジュールをもとに、PM の停止スケジュールを作成する。対象の PM 内の VM を所持している全てのユーザの停止スケジュールを 15 分ごとに参照し、対象の 15 分が、PM 内の VM を所持している全てのユーザの停止スケジュールにおいて停止状態と判断されている場合、停止スケジュールに追加する。これを対象となる 7 台の PM すべてに対して行う。

5. 評価実験

評価実験では提案を適用して、PM の起動、停止を行うスケジュールを作成し、その PM 停止スケジュールを PM 停止スケジュールの正解率で評価した。PM 停止スケジュールを作成するのに使用したデータの収集期間である 4 週間の次の 1 週間のユーザの PM 内の VM 上の OS へのアクセスと、作成した PM 停止スケジュールを比較する。比較の際、PM における未使用時間に占める停止時間の割合、PM ごとの削減可能な電力量、PM の停止時間中における VM 使用時間の割合の 3 つの項目で評価する。

PM の未使用時間に占める停止時間の割合は、PM 内の VM で作業が行われていなかった時間のうち、提案アルゴリズムによって PM を停止する事ができた時間の割合を示す。

PM ごとの削減可能な電力量は、PM の停止スケジュール中の停止時間のうち、ユーザからの PM 内の VM 上の OS へのアクセスがなく実際に停止が可能である時間に、PM を停止した場合に削減可能な電力量を示す。PM ごとの電力消費量の平均値に、24 時間のうちの PM の停止スケジュール中の停止時間のうち、ユーザからのアクセスがなく実際に停止が可能である時間の割合を掛け合わせて求める。削減可能な電力量は kWh(キロワット時)で示す。

PM の停止時間中における VM 使用時間の割合とは、PM の停止スケジュール上では停止時間であったが、実際は

*2 <https://takeout.google.com/settings/takeout>(参照 2024/12/23)

ユーザからのアクセスがあった時間の割合を示している。

提案では過去 10 週間分の歩数のデータを使用予定だったが、実験期間が不足していたため過去 4 週間分のデータで実験を行った。また、歩数収集に使用した Google Fit から取得できるデータの最小粒度が 15 分だったため、ProBy-Access(t) と ProByStep(t) は 15 分間隔で求めている。

実験環境

CDSL で運用されている 7 台の PM である、Jasmine, Rose, Plum, Violet, Mint, Lotus, Lily を対象とする。PM には VMwareESXi が導入されていて、仮想化環境が構築されている。また、対象となる PM 内の全ての VM に、VM 使用状況のログを集積するツールを導入している。提案アルゴリズムを実行するツールは、VM 上の OS へのアクセス状況のログを集積する VM へ配置する。

実験結果と分析

PM ごとに提案アルゴリズムの評価を示す。まず、PM の未使用時間における停止時間の割合を評価する。図 9 は PM ごとの未使用時間における停止時間の割合を示している。

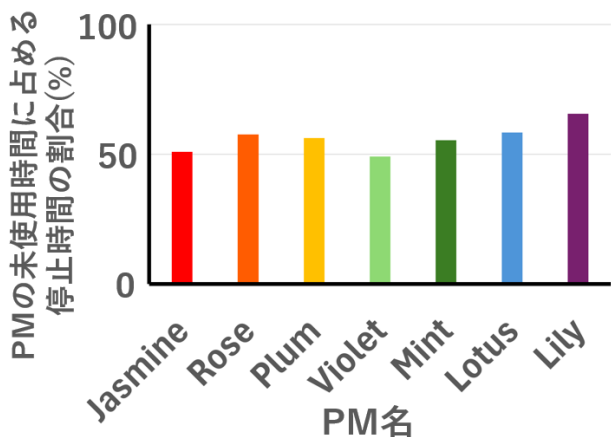


図 9 PM の未使用時間における停止時間の割合

PM ごとの未使用時間における停止時間の割合は、1 週間の PM の未使用時間における、停止時間の割合の平均を示す。X 軸は対象となった 7 台の PM の名称を表し、Y 軸は未使用時間における停止時間の割合を表している。提案アルゴリズムを適用した結果、対象とした 7 台の PM 全体の、PM が使用されていなかった時間における停止時間の割合は約 55.22%であった。対象となる 7 台の PM の未使用時間における停止時間の割合は、Jasmine は未使用時間が約 1438 分で、停止時間が約 733 分のため約 50.96%、Rose は未使用時間が約 1346 分で、停止時間が約 775 分のため約 57.61%、Plum は未使用時間が約 1440 分で、停止時間が約 810 分のため約 56.25%、Violet は未使用時間が約 1416 分で、停止時間が約 697 分のため約 49.19%、Mint

は未使用時間が約 1438 分で、停止時間が約 797 分のため約 55.44%、Lotus は未使用時間が約 1438 分で、停止時間が約 840 分のため約 58.40%、Lily は未使用時間が約 1393 分で、停止時間が約 915 分のため約 65.66%である。

7 台の PM の中で、PM の未使用時間における停止時間の割合の平均が最も低かった PM は、Violet で約 49.19%であり、割合が最も高かった PM は Lily で約 65.66%であった。未使用時間における停止時間の割合が十分に向上しなかった理由は、提案スケジュールの作成の対象とした期間は多くのユーザが基礎実験を行っていたが、評価実験を行った期間はレポートの提出のためにレポートを作成しており、VM での作業を行っていなかったためである。

次に、停止スケジュールを適用した際の削減可能な電力量を評価する。

提案適用前の 1 日における PM ごとの消費電力量は、Jasmine が約 1.87kWh、Rose が約 2.95kWh、Plum が約 3.87kWh、Violet が約 3.14kWh、Mint が約 2.3kWh、Lotus が約 2.56kWh、Lily が約 3.92kWh である。提案適用後は、Jasmine が約 0.92kWh、Rose が約 1.36kWh、Plum が約 1.69kWh、Violet が約 1.62kWh、Mint が約 1.03kWh、Lotus が約 1.07kWh、Lily が約 1.43kWh となった。提案適用によって削減可能な電力の割合は、Jasmine が約 50.88%、Rose が約 53.84%、Plum が約 56.25%、Violet が約 48.39%、Mint が約 55.35%、Lotus が約 58.32%、Lily が約 63.50% である。提案適用後の消費電力量は、未使用時間中の停止時間が 1 日に占める割合を提案適用前の消費電力量に掛け合わせることで求めた。提案適用によって削減可能な電力の割合は、提案適用後の消費電力を提案適用前と提案適用後の消費電力量の差で割ることで求めた。

停止時間中にアクセスがあった割合を評価する。停止時間中にアクセスがあった割合は、対象とした 7 台の PM 全体では約 0.90 %である。

PM の停止時間中にアクセスがあった割合は、Jasmine が 0%、Rose が約 4.13%、Plum が 0%、Violet が約 1.25%、Mint が約 0.29%、Lotus が約 0.32%、Lily が約 0.30% である。Rose は約 4.13%と 7 台の PM で最も停止時間中にアクセスがあった割合が高い。この理由は、Rose は VM の利用者が多く、その分過去のアクセス時間を超えて利用するユーザが多いためである。

6. 議論

本稿の提案では、アクセス可能性を示す 2 つの値である ProByAccess と ProByStep を求める際に直近 10 週間分のデータを使用している。直近 10 週間分のデータを使用する理由として、人間の習慣の形成にかかる時間が平均 66 日かかることをあげている。しかし、この期間は新たな習慣を形成するためにかかる日数であり、すでに習慣化されている行動の習慣性を示す指標としては適切ではない。そ

のため本稿で提案した、直近 10 週間分のデータの使用は PM の停止スケジュールの作成において最適であるとは言えない。実際に収集した歩数のデータから期間を変化させて PM の停止スケジュールを作成し、それぞれの期間の PM の停止スケジュールの精度を比較することで PM の停止スケジュールの作成に最適な期間を求める。この時、できる限り短い期間で PM 停止スケジュールの精度が向上する期間がより最適な期間といえる。理由は、PM の停止スケジュールの作成に使用する、PM 内の VM 上の OS へのアクセスログとユーザの歩数のデータが少なく済むからである。

本稿の提案では、ユーザの歩行の有無を判断する時間の間隔を、ユーザの歩行の特徴の取得に最適な時間として 1 分間隔に設定している。ユーザの移動距離を測定し、収集した歩数のデータとユーザの移動距離を時間ごとに比較することで、ユーザの移動の有無を判断するのに最適な時間を求めることができる。例えば、歩行速度には個人差があり、ゆっくり歩く人と速く歩く人とは 1 分間に歩く歩数に差がある。ゆっくり歩く人が 1 分間に 30 歩進み、速く歩く人が同じ時間で 100 歩進む場合、移動距離の差は約 3 倍である。歩数のデータと移動距離の間に比例関係があるとすると、歩数のデータと移動距離を時間ごとにプロットすることで、移動の有無を示す閾値を求めることができる。

本稿の提案では、ユーザが歩行している時間帯を判断するために、時間 t におけるユーザの歩数の平均を使用している。ユーザが、平均歩数よりも多く歩いた時間帯をユーザの歩行がある時間帯と定義しているが、ユーザの歩行がある時間帯の定義を見直すことで、ユーザが作業を行っていない時間をより正確にし、PM 停止スケジュールの精度が向上する。歩数のデータの散らばりを求め、ユーザごとの歩行のデータの偏りを把握することで、ユーザの歩行の有無を判断する最適な閾値を求めることができる。

本稿の提案では、ユーザの移動している時間帯を把握するためにユーザごとに歩数のデータを取得しているが、歩数のデータの規則性がユーザによって異なる。ユーザの歩行の規則性の強さを示す値を設定し、その値によって PM 停止スケジュールに及ぼす影響度を変化させることで、ユーザの規則性の強さを考慮した PM 停止スケジュールを作成することができる。これにより PM 停止スケジュールの精度が向上する。

本稿の提案では、ユーザが PM 内の VM 上の OS へアクセスする可能性を示す値である ProByAccess をユーザごとに設定しており、ユーザのアクセス可能性を PM ごとに区別していない。ユーザのアクセス可能性を示す ProByAccess を PM ごとに設定することで、停止スケジュールの精度が向上する。また、ある PM 内の VM 上の OS にユーザがアクセスした際に、同一ユーザの別 VM 上の OS にアクセスがあるかどうかを示す値を設定することで、ユーザのアク

セス可能な時間を影響させつつ、PM ごとのアクセス可能性から PM の停止スケジュールを作成することができる。

本稿の実験では、アクセス可能性を示す 2 つの値である ProByAccess と ProByStep を 15 分間隔で求め、実験を行っている。Google Fit を使用して収集できる歩数データは 15 分ごとであるため、本稿の実験でもアクセス可能性を示す値の設定を 15 分間隔とし、15 分ごとに PM の起動、停止の判断を行ったが、この値を異なる値に設定することで停止スケジュールの精度が向上する。Google Fit API を使用することで、本稿の提案と同じく 1 分ごとの歩数データの取得が可能になる。Google の認証プロセスを得て API を利用することで、本稿の実験よりも高頻度で歩数データを取得することができるため、最適な間隔で 2 つの値を求めることができる。

7. おわりに

東京工科大学コンピュータサイエンス学部の Cloud and Distributed Systems Laboratory には、PM が 10 台あり、CDSL に所属している学生が使用できる PM は 7 台ある。それらは学生が常に VM で作業を行えるよう、24 時間稼働している。課題は、ユーザが PM 内の VM 上の OS へアクセスしていない時間帯にも PM が起動しており、電力を消費していることである。提案では、ユーザの歩数データと PM 内の VM 上の OS へのアクセス時間のデータをもちいてユーザがアクセスする可能性を示す ProByAccess と、ユーザがアクセスしない可能性を示す ProByStep を設定し、2 つの値を比較することで、PM が使用されていない時間を予測し、PM の停止スケジュールを作成し、PM の停止、起動を行うことで PM の消費電力を削減する手法を提案する。評価実験では、提案したアルゴリズムをもちいて PM の停止スケジュールを作成し、停止スケジュールの正解率を、PM が使用されていない時間に停止可能な時間、削減可能な電力量、停止した時間にユーザが作業した時間の 3 点で評価する。評価は作成した停止スケジュールと 1 週間のユーザからのアクセスを比較し、スケジュールの正解率を計算することで行った。停止スケジュールを適用することで、結果として、PM が未使用の時間を最大で約 65.66%、最小で約 49.19% 停止できた。停止すると予測した時間のうち、約 0.90% の時間には VM が使用されていた。電力では各 PM の平均で電力消費の約 56.21% の電力を削減をすることができた。

参考文献

- [1] Çelik, D., Meral, M. E. and Waseem, M.: A New Area Towards to Digitalization of Energy Systems: Enablers, Challenges and Solutions, *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6 (online), DOI: 10.1109/ECAI54874.2022.9847313 (2022).

- [2] Zhang, L., Zhuo, Y., Wang, T., Wang, T. and Chen, J.: Application of Data Center Technology in Power Data Resource Sharing, *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, pp. 1406–1411 (online), DOI: 10.1109/ICDSCA59871.2023.10392731 (2023).
- [3] Hogade, N., Pasricha, S. and Siegel, H. J.: Energy and Network Aware Workload Management for Geographically Distributed Data Centers, *IEEE Transactions on Sustainable Computing*, Vol. 7, No. 2, pp. 400–413 (online), DOI: 10.1109/TSUSC.2021.3086087 (2022).
- [4] Hirose, K.: DC powered data center with 200 kW PV panels, *2018 International Power Electronics Conference (IPEC-Niigata 2018 -ECCE Asia)*, pp. 822–825 (online), DOI: 10.23919/IPEC.2018.8507890 (2018).
- [5] Garg, N., Singh, D. and Goraya, M. S.: Power and Resource-Aware VM Placement in Cloud Environment, *2018 IEEE 8th International Advance Computing Conference (IACC)*, pp. 113–118 (online), DOI: 10.1109/IADCC.2018.8692118 (2018).
- [6] Ranganathan, P., Leech, P., Irwin, D. and Chase, J.: Ensemble-level Power Management for Dense Blade Servers, *33rd International Symposium on Computer Architecture (ISCA'06)*, pp. 66–77 (online), DOI: 10.1109/ISCA.2006.20 (2006).
- [7] Kochupurackal Rajan, S., Ramakrishnan, B., Alissa, H., Kim, W., Belady, C. and Bakir, M. S.: Integrated Silicon Microfluidic Cooling of a High-Power Overclocked CPU for Efficient Thermal Management, *IEEE Access*, Vol. 10, pp. 59259–59269 (online), DOI: 10.1109/ACCESS.2022.3179387 (2022).
- [8] Pinheiro, E. and Bianchini, R.: Author retrospective on energy conservation techniques for disk array-based servers, *ACM International Conference on Supercomputing 25th Anniversary Volume*, New York, NY, USA, Association for Computing Machinery, p. 71–73 (online), DOI: 10.1145/2591635.2591666 (2014).
- [9] Pinheiro, E. and Bianchini, R.: Energy conservation techniques for disk array-based servers, *Proceedings of the 18th Annual International Conference on Supercomputing*, ICS '04, New York, NY, USA, Association for Computing Machinery, p. 68–78 (online), DOI: 10.1145/1006209.1006220 (2004).
- [10] Lally, P., Van Jaarsveld, C. H. M., Potts, H. W. W. and Wardle, J.: How are habits formed: Modelling habit formation in the real world, *European Journal of Social Psychology*, Vol. 40, No. 6, pp. 998–1009 (online), DOI: 10.1002/ejsp.674 (2010).