

ネットワークシステムの省電力化のためのプランニング手法 AI Planning Method for Energy-efficient Network Systems

栗原 孝太[†]
Kouta Awahara

和泉 諭[‡]
Satoru Izumi

阿部 亨[§]
Toru Abe

菅沼 拓夫[§]
Takuo Suganuma

1. はじめに

オフィスビル等で消費される電力のうち、ネットワークシステムによる消費電力が占める割合は益々増大してきており、ICT のグリーン化が喫緊の課題となっている。従来のネットワークシステムのグリーン化では、個々の ICT 機器の省電力化の取り組みが積極的に行われてきた。一方、サーバ、ネットワーク機器、空調などを含む系全体の挙動に配慮して運用を制御することで、より高い消費電力低減効果が得られると考えられるが、その技術は未だ確立されていない。

我々は、ネットワークシステムを構成する各機器、具体的にはサーバ、ネットワークスイッチ、およびこれらの機器の稼働に必要な空調設備などを統合的かつ自律的に制御し、電力の有効利用を実現する手法を提案している [1]。本手法は大きく最適化とプランニングの 2 ステップから構成されている。本研究では、その中のプランニング機構の設計および実装について詳述し、実験を通して本プランニング手法の有効性を確認する。

2. 関連研究

ネットワークシステムにおける省電力化を行う手法として、データセンターの省電力運用管理手法が提案されている [2]。この手法では、予め各サーバに設定した重要度に従って処理を実行させるため、状況に応じた柔軟な割り当てが困難であるという課題がある。また、クラウド環境に特化したサーバ資源制御の手法が提案されている [3]。この手法では、システムを構成する機器のうちサーバのみに着目しており、ネットワーク機器など他の機器を考慮に入れていないという課題がある。

3. 提案手法の概要

サーバだけでなくネットワーク機器や空調も含め統合的に制御する際、系全体が複雑になり、適切な手順で機器を制御することが必要になる。そこで、本手法では AI 技術の一つである「プランニング」を用いる。プランニングとは「与えられた目標を達成するために必要な行為の系列を自動生成する」ことであり、これを用いることで、ネットワーク機器や空調を含むシステムの構成変更の手順を自動生成することができる。

提案するプランニング手法を用いたシステムの概略を図 1 に示す。ネットワーク内に Operation Server を導入し、全体の制御を行う。まず、各サーバの使用率

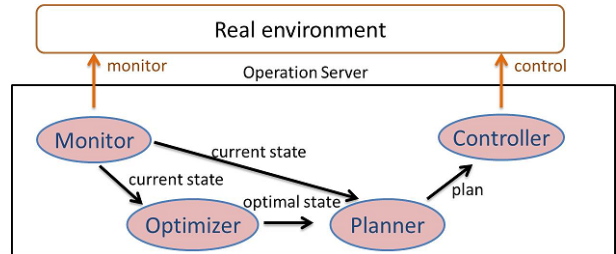


図 1: プランニング手法を用いたシステム概略図

- oper (アクションの名前)
vars: (関係するオブジェクト)
pre: (前提条件)
del: (削除するオブジェクトorステート)
add: (追加するオブジェクトorステート)
- oper moveTask [タスクを移動する]
vars: ?fromserver, ?toserver, ?t
pre: dom:ServerOn(?fromserver),
dom:ServerOn(?toserver),
dom:ProcessingTask(?t),
dom:hasTask(?fromserver, ?t)
del: dom:hasTask(?fromserver, ?t)
add: dom:hasTask(?toserver, ?t)
- oper turnServerOn [サーバをオン]
vars: ?server, ?switch
pre: dom:ServerOff(?server),
dom:hasOrder(?switch, ?server),
dom:SwitchOn(?switch)
del: dom:ServerOff(?server)
add: dom:ServerOn(?server)
- oper turnServerOff [サーバをオフ]
vars: ?server, ?switch
pre: dom:ServerOn(?server),
dom:hasOrder(?switch, ?server),
dom:SwitchOn(?switch)
del: dom:ServerOn(?server)
add: dom:ServerOff(?server)

図 2: アクションの構文規則と記述例

を観測する (Monitor)。次にシステム全体の消費電力が最少となるタスクの割り当て方 (最適状態) を求める (Optimizer)。これを基に、現在の状態から最適状態へと移行するプランを生成する (Planner)。最後に、実際にネットワーク機器、空調設備のオン/オフや設定変更を行う (Controller)。本稿ではこのうち、特に Planner の設計・実装について詳しく述べる。

4. 設計・実装

本研究では、Planner の設計・実装に DLPlan[4] を用いる。DLPlan の構文規則に基づき、ネットワークシステムを構成する各機器やサーバ上で実行されるタスクをオブジェクトとして定義した。表 1 に各オブジェクトの定義を示す。また、DLPlan におけるアクションの構文規則と、それに基づくアクションの記述例の一部を図 2 に示す。ここでは、スイッチをオン/オフにするアクション turnSwitchOn/Off や、空調をオン/オフにする turnACOn/Off など定義している。

表 1: オブジェクトの定義

ServerOn	電源が ON のサーバ
ServerOff	電源が OFF のサーバ
SwitchOn	電源が ON のスイッチ
SwitchOff	電源が OFF のスイッチ
ACOn	電源が ON の空調
ACOff	電源が OFF の空調
FreeTask	サーバに割り当てられる前のタスク
ProcessingTask	サーバに割り当てられ実行中のタスク

[†]東北大学大学院 情報科学研究科, Graduate School of Information Sciences, Tohoku University

[‡]東北大学電気通信研究所, Research Institute of Electrical Communication, Tohoku University

[§]東北大学 サイバーサイエンスセンター / 大学院情報科学研究科, Cyberscience Center / Graduate School of Information Sciences, Tohoku University

5. 実験

5.1. 実験の概要

予備実験として、Planner の性能評価を行った。本実験の実行環境を表 2 に示す。例として、図 3 に示すサーバ 4 台、ネットワークスイッチ 2 台、空調 2 基からなるネットワークシステムで 2 つのタスクが処理されている環境を想定しシミュレーション実験を行った。図 3 の左側を初期状態とし、右側を目的状態とすると、各状態は図 4 のように表現できる。

表 2: 実行環境

PC	Dell Optiplex 980
OS	Windows 7 Professional 64bit
CPU	Intel(R) Core(TM) i7 870 @ 2.93GHz
RAM	4.00GB
Planner	DLPlan

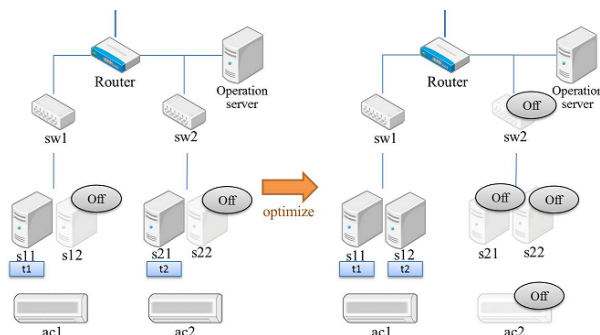


図 3: ネットワークシステムにおけるプランニングの例

- initial state
 - dom:ServerOn(prob:s11),
 - dom:ServerOff(prob:s12),
 - dom:ServerOn(prob:s21),
 - dom:ServerOff(prob:s22),
 - dom:SwitchOn(prob:sw1),
 - dom:SwitchOn(prob:sw2),
 - dom:ACOn(prob:ac1),
 - dom:ACOn(prob:ac2),
 - dom:ProcessingTask(prob: t1),
 - dom:ProcessingTask(prob: t2),
 - dom:hasTask(prob:s11; prob: t1),
 - dom:hasTask(prob:s21; prob: t2),
 - dom:hasOrder(prob:sw1; prob:s11),
 - dom:hasOrder(prob:sw1; prob:s12),
 - dom:hasOrder(prob:sw2; prob:s21),
 - dom:hasOrder(prob:sw2; prob:s22),
 - dom:hasOrder(prob:ac1; prob:sw1),
 - dom:hasOrder(prob:ac2; prob:sw2)
- goal state
 - dom:ServerOn(prob:s11),
 - dom:ServerOn(prob:s12),
 - dom:ServerOff(prob:s21),
 - dom:ServerOff(prob:s22),
 - dom:SwitchOn(prob:sw1),
 - dom:SwitchOff(prob:sw2),
 - dom:ACOn(prob:ac1),
 - dom:ACOff(prob:ac2),
 - dom:ProcessingTask(prob: t1),
 - dom:ProcessingTask(prob: t2),
 - dom:hasTask(prob:s11; prob: t1),
 - dom:hasTask(prob:s12; prob: t2),
 - dom:hasOrder(prob:sw1; prob:s11),
 - dom:hasOrder(prob:sw1; prob:s12),
 - dom:hasOrder(prob:sw2; prob:s21),
 - dom:hasOrder(prob:sw2; prob:s22),
 - dom:hasOrder(prob:ac1; prob:sw1),
 - dom:hasOrder(prob:ac2; prob:sw2)

図 4: 実験における初期状態と目的状態

5.2. 実験結果および考察

初期状態と目的状態を入力として、DLPlan を用いてプランを生成した。出力されたプランを図 5 に示す。

ここでは、まずタスクの移動先のサーバ s12 をオンにしてからタスク t2 を移動、その後サーバ s21、ネットワークスイッチ sw2、空調設備 ac2 の順でオフにしている。このように、タスク、サーバ、ネットワークスイッチ、空調設備の制御が正しい順序でプランが生成されることがわかった。

```

11 time solving problem: 81408 ms
12 Total time solving problems: 81408 ms
13
14 Problem: problem1
15 Plan: [turnServerOn{?server=prob:s12, ?switch=prob:sw1},
16 moveTask{?t=prob:t2, ?fromserver=prob:s21, ?tserver=prob:s12},
17 turnServerOff{?server=prob:s21, ?switch=prob:sw2},
18 turnSwitchOff{?switch=prob:sw2, ?ac=prob:ac2},
19 turnACOff{?ac=prob:ac2}]
20
21

```

図 5: 生成されたプラン

今回の実験ではプランニングに約 1 分半の時間を要した。新たにタスクが追加されるたびにプランニングを行うことを想定しており、計算中に新たなタスクが追加されると新たなタスクを考慮して再度プランニングを行う必要が発生するため、計算時間は短いことが求められる。そこで、プランニングを二段階に分割し、階層的に実行することで計算時間の短縮を試みた。先ほどの実験と同様の問題について空調の制御とそれ以外の操作に分割しプランニングを行った。この結果を図 6 に示す。図 6 に示すとおり、計算時間を約 8 秒にまで短縮することができた。

```

17 time solving problem: 14 ms
18 Total time solving problems: 7845 ms
19
20 Problem: problem1
21 Plan: [turnServerOn{?switch=prob:sw1, ?server=prob:s12},
22 moveTask{?t=prob:t2, ?fromserver=prob:s21, ?tserver=prob:s12},
23 turnServerOff{?switch=prob:sw2, ?server=prob:s21},
24 turnSwitchOff{?switch=prob:sw2, ?ac=prob:ac2}]
25
26 Problem: problem2
27 Plan: [turnACOff{?ac=prob:ac2}]
28
29

```

図 6: 追加実験で生成されたプラン

6. おわりに

本稿では、ネットワークシステム全体の省電力化を行うため、消費電力最適化のためのプランニング手法をネットワークシステムに応用し、システムを構成する各機器の自律的な制御を行う手法を提案した。特にオープンソースのプランニングシステムを用いた実装について詳しく述べた。また、実験を通してプランニングの有効性を確認し、実用化に関する検討を行った。今後は、システム全体の実装を行い提案手法の有用性を検証する予定である。

謝辞 本研究の一部は、総務省 SCOPE(旧 PREDICT)「情報システムの省電力化を実現する次世代ネットワーク管理技術の研究開発」の支援を受けて実施している。

参考文献

- [1] 栗原孝太他, “ネットワークシステムにおける電力の有効利用のための自律的制御手法,” 情報処理学会第 75 回全国大会, Vol.3, pp. 89-90, 2013.
- [2] Daisuke Arai, et al. “Proposal on Eco-friendly Operation Scheme for Reducing Energy Consumption of Data Center,” Network Operations and Management Symposium (AP-NOMS) 13th Asia-Pacific, pp.1-7, 2011.
- [3] Young Choon Lee, et al. “Energy efficient utilization of resources in cloud computing systems,” The Journal of Supercomputing, Volume 60, Issue 2, pp 268-280, May 2012.
- [4] Antonio A. Sánchez-Ruiz, et al. “DLPlan,” <http://dlplan.sourceforge.net/>, 2008.