

## 演習用WS上の無為プロセスの探索

永松 礼夫<sup>1</sup> 諏訪 大介<sup>2</sup>

教育・研究のどちらの側面からも、計算機群の効率的な管理は重要である。本学の演習室のワークステーションの使用状況を調べていると、ユーザが誰もいないのにかなりの負荷が観測されるホストが見られるが、これは無為に実行されているプロセスが原因である。教育の側面からは、これらプロセスの所有者は警告されるべきである。研究の側面からは、ホストの負荷挙動を追跡することは分散処理に有用である。本学には演習室が複数あり、ホストの総数はかなりの数になるが、それらすべてのホストについて無駄に実行されているプロセスを調べる方法について研究した。本論文では、まず、「無駄に実行されている」プロセスの判定基準を提案しそれに基づいてプロセスを調査・分類する実験を行なった結果について報告する。

Searching Idly Running Processes  
on Workstations in ClassroomsLeo Nagamatsu<sup>1</sup> Daisuke Suwa<sup>2</sup>

To establish an effective management system for computer cluster is important, both from educational and research viewpoint. In classroom workstations of our university, some hosts are heavily loaded when no one has logged in. This load is mainly caused by idly running process. In educational situation, owner of these processes must be warned. In research view, tracing load behavior of each host is useful for distributed computing. The total number of classroom workstations is quite large, with many educational computer classrooms of our university. We studied to search idly running processes, and then report the result of searching experiments for those idly running processes.

## 1 はじめに

近年、学生が自主的に学習できる環境を提供するためなどの理由から、大学の演習室に多数のワークステーション群 [1, 2] が導入されるケースが多く見られる。これらを分散計算の実験場所として用いることは魅力的であり、例として、同じ大学内の演習室のすべてのワークステーションが協調して一つの問題を数日かけて解く、といった状況が想定できる。

しかし、利用できる計算資源は均等ではなく、数時間以上といった長い範囲で見ると、あるホストはダウンしている可能性があるし、通信やファイル・マウントが渋滞したり失敗する可能性もある。また、コンソールの前に座ったユーザのプロセスを優先させる、といった条件も考慮したい。

このような状況の下で、各ホストにどうすれば「効果的」に仕事を分配する [3] かは自明ではない。異なる機種が混在する環境では、各ホストに応じた仕事の定量的管

理 [4] が要求される。また、同じ機種でも負荷の少ないホストよりも負荷の多いホストでは仕事の処理は遅いし、ダウンにより処理ができないホストを考慮する必要もある。その実現には、負荷の少ないホストに優先的に仕事をあたえ、負荷の多いホストやダウンしているホストは避ける判断が必要である。その前提として各ホストが現在どのような状況なのか把握し、適切な対策をすること [5] も必要である。

予備調査として、ホストの使用状況を調査していると、かなりの負荷がみられるが、誰もユーザがいなかったホストが見られる。本学においては、長時間計算を意図的にバックグラウンドで処理させるユーザは意外に少なく、この殆んどは無駄に実行されているプロセスによる。ユーザの不注意で、ログアウト時に kill されなかったエディタやブラウザ関連のプロセスで、あるものは空転走行を続けて CPU を浪費している。さらに、優先度を下げる設定がなされていないことが多く、他ユーザの処理に影響することが多い。これらの無為プロセスの存在により、CPU 資源が無駄に使われ、メモリが無駄に占有されている。

本研究では、教育機関において計算機群を管理する立

<sup>1</sup>(nag@u-aizu.ac.jp) 会津大学 情報センター, ISTC, The University of AIZU, Aizu-wakamatsu, Fukushima, 965-8580 JAPAN

<sup>2</sup>(m5021115@u-aizu.ac.jp) 会津大学 大学院, Graduate School of Computer Science and Engineering, The University of AIZU

場より、それら「迷惑なプロセス」に早期に対応するため、それらを探し、判別する方法を提案し、実際に探索を行なうプログラムを試作した。また、本学の演習室での無為に実行されているプロセスの探索結果について述べる。

## 2 背景

### 2.1 無為プロセス

無為プロセスを、残存プロセスと空転プロセスに分類して扱うこととした。

#### 2.1.1 残存プロセス

典型的な無駄なプロセスとして、オーナがログアウトした後もそのホストに残っているプロセスがある。それを残存プロセスと呼ぶことにし、control terminalがなく(ユーザとの対話セッションが張られてない)、オーナはそのプロセスの存在しているホストにログインしていないプロセスとする。

デーモンプロセス しかし、デーモンプロセスとして、意図的に残しているものを残存プロセスとして認識してしまうという問題点がある。例えば、本学においては seekd プロセス(ユーザがどのホストにログインしているか探すツール)がある。これを防ぐためには、ユーザから申告してもらい、区別しなければならない。

遠隔起動 リモートシェルによって他のホストからコマンドの実行ができる。そのうち、rsh コマンドで実行されたものは、システムが in.rsh や rshd プロセスなどの媒介プロセスを生成し、その子プロセスとして当該プロセスが実行されるので、親プロセスを調べれば区別することができる。しかし、xrsh コマンドで実行されたプロセスは in.rsh や rshd プロセスがないため区別することはできない。

#### 2.1.2 空転プロセス

他のホスト、特に各研究室のX端末などから、長時間ログインし続けるユーザがおり、そのユーザのプロセスが暴走している場合が多くある。それらを空転プロセスと呼ぶことにした。プロセスのオーナはそのプロセスのあるホストにログインしているが、プロセスは実行状態でCPU使用率が高く、かつ、control terminalがないものとする。

### 2.2 会津大学内の演習室

本学の特徴として、演習室のすべての計算機は UNIX ワークステーションである。また、ホストの総数は 406 台とかなりの数になる。本学における計算機演習室の部屋名、WS の設置台数、OS(Version)、利用可能時間は表 1 の通りである。そのうち全体の 76% が Sun OS であり、Ver 5.5.1(Solaris) と Ver 4.1.4 に約半分ずつ別れる。

演習室のワークステーションは、授業中での使用、授業時間外などの課題処理、空き時間の自由利用などに活用されている。主に Computer Exercise Room(CER) は一般用で literacy, programming language, algorithms, Database systems などの授業で使用されている。Hardware Workshop は CAD による回路設計ソフトウェアが利用できるため、logic circuit design, computer architecture などの授業で使用されている。CAI Room は語学関連の technical writing, composition などの授業で使用している。また、CER 3,4 は最新機種、CER 5,6 は充実したグラフィック環境という理由から人気がある。

## 3 無為プロセスの探索

### 3.1 探索方式

残存および空転プロセス探すプログラムを perl スクリプト(6 ファイル、計 385 行)として実現した。一台のホストから、他のホストを一台ずつ逐次的に、リモートシェルでプロセス探索スクリプトを実行する方式で、全ホストのプロセス情報の収集を行なう。被探索ホスト上では、プロセス状態の一覧を得る ps コマンドと、ログインしているユーザを調べるコマンドのみが実行され、その結果を一台のホストに集めて 図 1 の判定フローに基づいて処理する。

残存プロセス 存在するプロセスのうちで、オーナがログインしておらず、control terminal がないものを選択する。なお、システム(root user)が実行しているものは除外。

空転プロセス 存在するプロセスのうちで、オーナがログインしているものを選択する。そこから、実行状態(ps コマンドで得られる CPU 使用率が基準以上)で control terminal がないものを選ぶ。

判定法の問題点 xrsh コマンドによるプロセスはそれぞれの OS において in.rshd, rshd プロセスを持たないため、リモートシェルによって意図的に実行されているプロセスも残存プロセスと判定される。学生は昼間に授業があるので、昼間のほうが夜間や早朝よりログインしている人が多い。そこで、夜間や早朝に xrsh コマンドを実

表 1: 学内の演習室

Room name	number of host	OS(Version)	open time
CER 1-2(std1-2)	45 × 2	Sun OS(Ver.4.1.4)	24 hours
CER 3-4(std3-4)	51 × 2	Sun OS(Ver.5.5.1)	3: 24 h /4: 8:30-5:30
CER 5-6(std5-6)	49 × 2	IRIX(Ver.6.3)	5: 24 h /6: 8:30-5:30
HW 1-2(hdw1-2)	25 × 2	Sun OS(Ver.5.5.1)	24 hours
CAI Room 1-2 (cail-2)	33 × 2	Sun OS(Ver.4.1.4)	8:30-5:30

CER:Computer Exercise Room, HW: Hardware Workshop

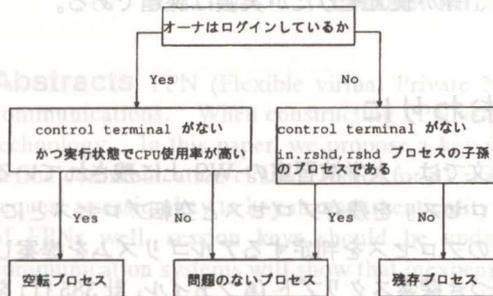


図 1: 残存・空転プロセスの判定方法

行しているユーザは少ないと考え、消極的措置ではあるが、このプログラムは夜間や早朝に実行した。

### 3.2 実験

#### 3.2.1 発見された無為プロセスの分類

一回の探査で発見された残存プロセスの種類は表 2 の通りである。プロセス総数 312 のうちそれぞれ CER1-2 で 54, CER3-4 で 102, CER5-6 で 74, hdw1-2 で 73, cail-2 で 9 が見つかった。

表 2: 残存プロセスの種類

ゾンビプロセス	74	(全室に分布)
cadence	43	回路 CAD(hdw のみ)
ウィンドウマネージャ	35	(全室に分布)
netscape	18	ブラウザ (全室に分布)
capture	29	ビデオカメラ (std 5-6 のみ)
a.out	8	実行ファイル (全室に分布)
mule	6	エディタ (全室に分布)
gs	5	ビューア (全室に分布)
その他	94	
総数	312	

また、空転プロセスは長時間ログインし続けた人のいる繁忙期にしか存在しなかった。

#### 3.2.2 高負荷ホストと無為プロセスの関係

プログラムを一度実行し、残存プロセスが発見された数は 312 であった。調査直前に各ホストの 1 分間のロードアベレージを調べてみると、1.00 未満のホストが 339

台であり、1.00 以上のホストが 67 台あった。また、残存プロセスが発見されたホスト数は 119 台であり、そのうち 1 分間のロードアベレージが 1.00 を越えていたホストは 57 台、202 プロセスが存在した。1 分間のロードアベレージが 1.00 を越えていたホストの 85% で全体の 65% の残存プロセスが発見された。

表 3: 負荷と残存プロセスの関係

	負荷>1.00	1.00>負荷	計
ホスト台数	67	339	406
残存プロセス有	57	62	119
残存プロセス総	202	110	312

#### 3.2.3 探索のオーバーヘッド

探索を起動する側のホストについて、負荷の時間変化を図 2 に示す。縦軸は負荷 (過去 1 分のロードアベレージ, 1 分ごと)、横軸は時刻 (分単位) で、プログラム開始は横軸の 30 の時点であり、実行時間は約 1 時間程 (横軸の 90 付近まで) であった。負荷は最大でも 0.3 程度であり、この実行は探索側ホストに負担となるものではないと言える。

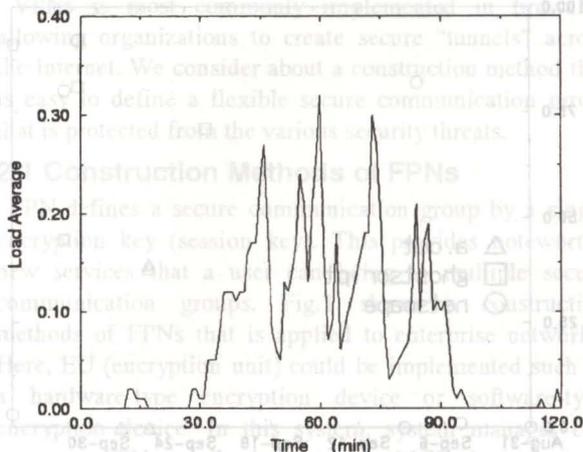


図 2: 探索側のホストの負荷変化

また、被探索ホストにおいて、1分間のロードアベレージを調査前と後で調べた。変化の差は平均で約 0.05 であった。前述のように、被探索ホスト上では、プロセス状態と現在ログイン中のユーザをだけを調べる方式としたため、影響を少なくできた。

#### 4 考察

実用的なオーバーヘッドで残存プロセスを調査できたが、さらに効率的にするために高負荷のホストに絞って調査することも考えられる。CPU 資源への影響だけを心配するならば、負荷値の高いホストのみを対策すればよい。台数にして 16%のホストを調べれば、プロセス数にして 65%の問題プロセスが発見できるからである。

また、図 3 は CER 1-2 において見つかった残存プロセスのうち、gs, netscape, a.out について、平均プロセッサ使用率(累積 CPU 時間を経過時間で割ったもの)とプロセスの寿命(プロセス開始時刻で表示)を示したものである。△が a.out(一般のユーザ作成プログラム)、□が gs(ghostscript:PSビューア)、○が netscape(www ブラウザ)のプロセスである。

この図で左上にいくほど、長時間無為に実行され続けたプロセスである。下にあるものは CPU 時間を使っていない。図より、上方にいくほど、他のユーザに迷惑をかけるプロセスであるということ、残存プロセスは CPU 使用率の低い(横軸付近にある)ものと CPU 使用率のかなり高い(70%以上)に大別できることがわかる。(中央の重なった△は同一ホストなので、得られる CPU を大部分使っている)。従って、管理者の立場では、CPU 使用率のかなり高いプロセス群のみに迅速に対応すればいいであろう。その意味で、負荷の低いホストを頻りに調査する必要はないと思われる。

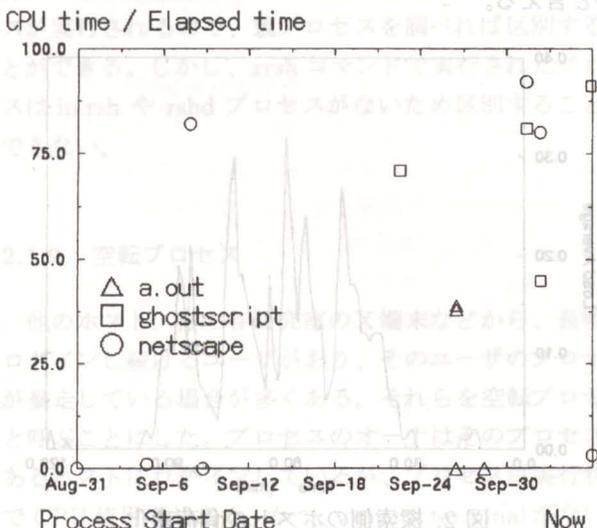


図 3: 残存プロセスの CPU 使用率と走行時間

なお、空転プロセスという分類を設けたことはあまり妥当ではなかった。同じ状況のプロセスが、オーナーがログインしているのは空転プロセスとなり、ログインしていない時は残存プロセスとなるからである。

また、本方式で区別できないケースとして `xrsh` コマンドによるプロセスがある。これを区別する指針として、累積 CPU 時間が短い、プロセッサ使用率が低い または優先度が低く設定されている、DISPLAY 環境変数が他のホストに設定されている、network 接続が別のホストにある、等が使えるのだが実装は課題である。

#### 5 おわりに

本論文では、大学演習室の WS 上に残されている「無為なプロセス」を残存プロセスと空転プロセスとに分け、それらのプロセスを判定するアルゴリズムを提案し、それに基づき探索スクリプト(6ファイル, 計 385行)を作成した。実際に演習室の計算機群を探索するには、406 台のホストを逐次的に走査する方式で 1 時間程を要した。

非特権ユーザが利用可能なコマンドのみで、軽いシステムを目指して構築したため、一部のプロセスについて十分に判別できないが、本方式は、管理者が学内の演習室ホストに存在する残存プロセスを探すのに有用であることが示された。

#### 参考文献

- [1] 石田純一 他, 室蘭工業大学における情報メディア教育システムの導入について, 情報処理教育研究会集会講演論文集, pp.550-553, 1997.
- [2] 堀勝博, 120 名まで同時利用可能な教育用計算機システムの構築, 情報処理教育研究会集会講演論文集, pp.554-557, 1997.
- [3] F.Tandialy, S.C.Kothari, A.Dixit, E. W. Anderson, **Batrun: Utilizing Idle Workstations for Large-Scale Computing**, IEEE Parallel & Distributed Thchnology, Vol.4, No.2, pp.41-48, 1996.
- [4] 大森洋一, 定量的なプロセッサ間通信を意識した並列化コンパイラ中間表現の提案, 情報処理学会, プログラミング研究会報告, 97-PRO-12(12-6), pp.33-37, 1997.
- [5] 森良哉 他, **PC サーバ・クラスタ — アベイラビリティ(可用性)向上をねらって —**, 情報処理, Vol 39, No.1, pp.49-54, 1998.