

デジタル Digital エッセイ

「そこはシュッと来てバーン」

坂口 裕靖

GPUのプログラミングとかそうなんですけど、多次元のデータを扱う場合、「早い方向」と「遅い方向」があります。仮に単純な足し算をして平均値を出すとして、早い方向に辿っていった場合と、遅い方向に辿っていった場合とでは処理時間に数十倍の差が出てきたりします。で、とりあえず処理できればいいなら遅かろうが早かろうが動けば一発 OK ですが、これがまた一回処理しただけで済むデータなんて殆ど無いわけ。結局二回以上繰り返すことになるため、予め最適化しておくことが重要になってきます。これがなかなか大変で、「たぶんこっちの方向で辿ったほうが早いだろうから、ここの処理を変更して、こっちを先にやってからあっちをやる」とか考えつつ、

コードに落としてコンパイルして動かしてみるわけです。で結果を測定してみると、運が良ければ成功したか失敗したかがわかります。通常は目に見えた違いが分からず、どちらが前だかわからないまま、嗅覚を駆使して最適と思われることを進めていくことになります。

ここでやっている作業は、結局コンパイル前のソースコードを変更して、処理が完了するまでにかかる実行コストを計測し、変更が有効だったか無効だったか、あるいは悪化したかどうかを判断して、変更したコードを残すかどうか、残さないなら次に何をするか、という手順を淡々と繰り返すだけです。本当は処理アルゴリズム自体を効率的なものに変えられればそれがベスト

なんですけど、そもそも何が効率的かがわからないのだから変え方がわからない。結果試行錯誤してうまく行ったやつを採用するわけですが、取りうるパラメータの組み合わせの数はあつという間に膨大になり、全部試してみるのも気が滅入ります。仕方ないので野生のカンに従って、一番処理コスト（通常は処理時間ですけど、使用するメモリ量とか、電力量など、何を最適化すべきかは状況によって千差万別です）が低い処理を選んでいくしかありません。報われるかどうかは野生のカンに依存します。

結局これ、少なくとも変更可能な部分が有限少数に落とし込めるのであれば、組合せ最適化問題を解いていることになります。ならば人間様ではなく、機械に任せられ

ワンポイント パスワード

4137

さて、ご期待通り xv の車台番号は、ばっちりリコール対象となっていました。現時点で国土交通省のリコール検索からはひっかかりませんが、「リコール・改善対策の届出」ページから 11/16 付の項目として、リコール届出番号 4137 として確認することができます。また、subaru のサイトの方からも「リコール等情報対象車両検索」から、型式と車台番号を入れることで、やはりリコール届出番号 4137 の pdf を確認することができます。国土交通省のサイトから確認できる pdf には「自動車使用者及び自動車分解整備事業者に周知させるための措置」が記述しており、これによれば「改善実施済車には、運転席側ドア開口部のドアロックストライカー付近に No.4137 のステッカーを貼付する。」そうです。一方、両方のサイトで確認できる「改善箇所説明図」によれば、識別のために「右ストラットタワー前方に黄色ペイントを塗布する。」とのことですので、No.4137 の

ステッカーおよび黄色いペイントが目印になるのでしょう。ところでこれ、国土交通省も subaru のサイトも、英語版のウェブサイトにはリコール情報にたどり着けるようなリンクが掲載されていないようです。まあ subaru の方はシェアを考えると日本在住の英語話者ユーザまで追っかけきれないのはわからなくてもいいですけど、所轄官庁の方はなんかケアした方が良さそうな気がします。そもそも国土交通省・自動車局・審査・リコール課 (<http://www.mlit.go.jp/jidosha/carinf/rol/>) のページに英語版のボタンないです。国内販売 150 万台前後の状態、今回のリコールでは約 37 万 6000 台が対象とのことですので、年間販売数の 25% 程度を検査する計算。このスケジュール調整がどれくらい大変なのか不明ですが、なにぶん人生初リコールなので、ちょっとドキドキしているところです。今のところはがきは届いていませんが、果たしていつ頃連絡が来るでしょうね？

ばべんりじゃないでしょうか。人間さまとは異なり、機械は気が減入ということがありませんから、膨大な組み合わせを淡々と試してみることができます。それでも時間の制約があるため、すべての組み合わせは調べきれないでしょうが、少なくとも統計的にベターな選択肢を選ぶことを期待できるのではないのでしょうか。もっといえば、この野生のカンをアルゴリズムに落とせるなら、そこそこのところまでは行けそうな気がします。さらにそもそも毎フレーム処理するようなものであるなら、そのまま最終プロダクトまで試行錯誤させて、運用中に進化するという期待ができるかもしれません。実際、最近の光学機器の性能を見れば、この機械頑張り作戦が有効に作用することがわかつています。そもそも最近の飛行機、機械頑張り以外で形決めてるとは思えないですし。

光学設計などの場合は画角とか収差とか面数上限とか組み付けの許容誤差などの制約条件内で最良の曲率・焦点距離・面数といったパラメータを決め込んでいくわけです。従来型の設計支援では光学設計や空力設計といったワン 이슈の最適化だったわけですが、最近のディーブリングを見てると、うまくネットワークに落とすことができれば、なんでも最適化できそうに見えます。さらに素晴らしいのが、「どうやって上手くいってるのか理解できないけど、なんかうまくいく」という特性でしょうか。言語化できない野生のカンを移植できる可能性があるのは、このふにゃふにゃしてる期待度にあります。

ここで、仮に任意のパラメータを元に、与えられた制約条件の中でベストなパラメータの組合せを探索するようなシステムが手軽に安価に使えるようになったとしましょう。ここでいう「安価」とは本当に安価であって、現在のビデオボードにおけるピクセル数あたりのコストと同様程度のものすごい「安価」だとします。ちなみに4K対応のビデオボードが仮に2万円だとする

と、ピクセル単価は約0.0023円です。この値段で一つのパラメータを一つの制約条件で最適化できると仮定すれば、最適化に対するイメージが大分変わってくるような気がしますませんか？

現在のコンピュータは、基本的にある仮想マシンをエミュレートする物理マシンという形態を取っています。Windowsの場合、WindowsのAPIを直接実行できる仮想マシン、Wがあるととして、その上で指定したアルゴリズムを実行するようにプログラムを組んでいます。ハイパーバイザはこれに仮想化の層を一つ増やすわけですが、同様の手順で最適化の層を組み込むことはそんなに珍しい考え方ではありません。すべてが仮想マシンの枠内で処理できれば幸せなのですが、そうは問屋がおろしません。最適化に手をつけようとする、ひとつ下のレイヤによる制約が露出するようになってきます。一度に使えるメモリは仮想レイヤのメモリ管理システムの制約を受けますし、処理時間は物理システムまでの全レイヤに依存します。そして、ゲームコンソールとは異なり、パソコンの実行環境は千差万別ですから、ある環境で最適な状態が任意のマシンで最適と仮定することができません。

さて、ここに物理マシンPの制約条件において、仮想マシンWを最適に動かすためのパラメータ調整をできる機構が挟まるとしましょう。そうすると、プログラムは仮想マシンWの範疇でのアルゴリズム最適化を行えば、物理マシンPの制約条件内なるべく安価に実行できるように機械が頑張ってくれそうです。考え方としては、一つのスレッドに一つの最適化機構が貼り付いて、物理マシンP上でより安価に実行できるようにパラメータを調整する感じでしょうか。速度モードで最適化するなら、消費電力や消費ストレージを気にすることなく、最短時間で処理が終了するよう調整しましょう。一方出先などで消費電力を極小にしたい場合、全く異なるパラメータの組合

せに落ち着くはずですが。これを実現するためには、そもそもの処理アルゴリズムを変える必要があるかも知れません。仮にアルゴリズムの選択が「これは犬」「これは猫」みたいな分類問題に帰着できるなら、最適化の範疇に収まる可能性が出てきます。一つの例としてはSQLをあげられるかも知れません。予め決めておいたテーブルの定義に従って、値の合計だの平均を取るのだといった抽象的な指示があって、実際の処理は最適化エンジンが物理的なデータの取得や処理を組み替えてなるべく低コストで実行できるように頑張る、という構造です。こんな感じでメタ・メタレベルのプログラムができるようになるのなら、実行時最適化を挟んでうまく稼働できるかも知れません。もっとも、そういった方向性で実用性のあるExcelクローンを作れるのかどうか、なんとも想像がつかませんが、きっと可能なんでしょう。

ここまでくると、そもそも仮想マシン相互の変換すら最適化でき、OSが事実上ユーザーから不可視の存在になってもよさそうです。実際bash on Ubuntu on Windowsとか、single page applicationとかみても、仮想化への志向は強いと思います。そりゃ苦労して作ったプログラムは骨の髄までしゃぶりいたいもんね。

仮に実行コスト最適化が機械の領分になるとすると、人間様のやることはざっくりした指示ということになるでしょう。「鉄人、頑張り」で制御できることを目指すのだとすると、ノリでは処理しきれないややこしい問題の大部分で破綻しそうな気がします。

Hiroyasu Sakaguchi
(株)IMAGICA イメージワークス