

IDEON-retreat レジюме オーバーレイと計算論
担当 doi 2006-04-03/04

本稿では、IDEON-retreat における「オーバーレイと計算論」における議論のタネとして、サーベイを行った結果を報告する。

前提として、doi は計算機科学の教育を部分的にしか受けておらず、MULTICS の講義は受けていても、今「生き残っていない」アーキテクチャについては知識ゼロの状態からスタートしている (和田先生が LISP マシンの話をするのを遠目に聞いていたことがあるくらい)。従って、本筋を外している部分も多々あるかもしれないが、ご容赦願いたい。(というか、以後散文体で書き殴ってます。ごめんなさい)

1 視点と背景

オーバーレイと計算論、という視点を持つに至った背景:

- doi:
昔から、オーバーレイ自身が環境を制御する、という漠然としたイメージを持っていた (多分 Sam の Neurogrid - 2001 年頃 - にインスパイア :p された)。ただし、ニューラルとか強化学習とかはイメージしっくりこない部分があった。
 - 今にして思えば、ニューラルも強化学習も理論体系がノイマンの枠内で語られるものしか表には残っておらず、非ノイマンな議論はディープな所に仕舞われていた (ような気がする)。
- Live E! Project の台頭
- 2005-12-20 ぐらい at IDEON meeting
キーワード: データフローマシン、データフロー的ソフトウェア (BOINC)、visual programming、関数型プログラミング、コネクションマシン、GHC
- 2005-12 WIDE 研究会 call for discussion
キーワード: 全世界のセンサノード、サーバ中心は嫌、うまい分散処理、定量的なスケラビリティの評価、データフローマシン、SIGMA-1、ノード抽象化
今思えばなんつーメタメタな議論かと
- その後、live E! で議論したりとか (図 1)。
- 2006-03 WIDE 研究会合宿 Live E! BoF: 担当教授陣の教育的指導により、僕だけ Live E! を追い出される :p

2 関連研究

- さまざまないにしえの計算機たち
- Iris: DHT 系の人々も合流してセンサーネットの話をしようとしている雰囲気。ただ、論文一覧見ると DB 屋の「ストリームに対する Query (CQ: Continuous Query)」みたいな研究が先行している。
- Hourglass(Harvard): データフローのオーバーレイを目指している。ノードの最適配置に情熱がある雰囲気 (別レジюме pietzuch-e... も参照)。

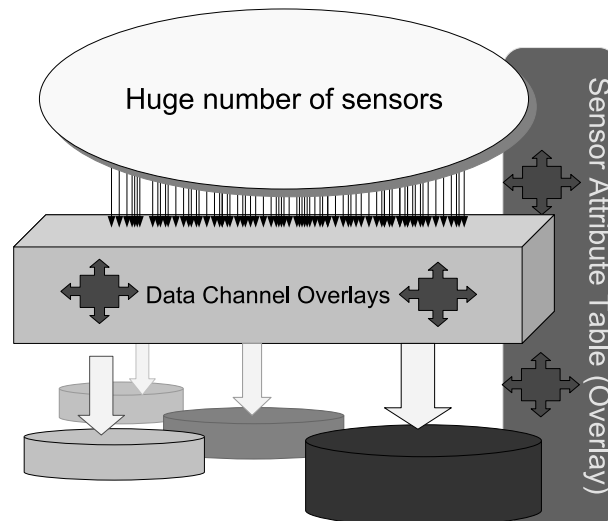


図 1: Live E! Overlay 概略

3 いにしえの計算機たち

Dennis, MIT 初代データフローマシン

静的データフローモデル。プログラムでループが表現できない。

SIGMA-1 128 プロセッサ動的データフローマシン。

平木, 「SIGMA-1: データフロースーパーコンピュータ」IPSJ Magazine Vol.43 No.2 Feb. 2002

Connection Machine ノイマン型分散計算機。これこそ Cell の先祖か？

ヒリス著, 喜連川訳 「コネクションマシン」パーソナルメディア社, 1990

他国内だけでも電総研とか NTT とか NEC とか ICOT とかいろいろなところがデータフローを含めている。 (とてもサーベイし切れず)

3.1 計算モデル

おおざっぱに分けて 4 つ (雨宮, 田中, 「コンピュータアーキテクチャ」, オーム社, 1988)。

命令型: ノイマン型はこれを下敷にしている。(等価かも)

命令と値を組として計算し、副作用という形でメモリに値を記録する。その値が望む計算結果となるように命令をくみたてる。命令の順序は強く束縛されている。

関数型: 計算の内容を関数の計算で表現する。純粋な関数型には副作用は存在しないので、関数の実行順序の束縛が弱く、並列化の抽出が容易という特徴がある。また、関数型はさらに適用型(計算)/単一代入型の二つに分かれる(意味的には等価なのかなこの二つ)。

論理型: 論理式を簡略化(reduction)していくことにより計算を行う。(正直、あまりよくわかっていません)

オブジェクト指向: 命令のかわりに主体(オブジェクト)を用意し、それらの間のメッセージのやりとりで計算を実行する。(これを計算モデルと思うのは気の迷いじゃないかと)

3.2 アーキテクチャ

ノイマン型: 命令型に対応するアーキテクチャ。命令とオペランドをメモリから読み出し、命令を実行し、結果をメモリ (レジスタ) に記録する、というサイクルの繰り返しで計算を実行する。

データフロー型: 計算のグラフを (メモリ内に) 構築し、データが各ノードに到着したもから任意の順序で計算を行う。グラフのノードが命令に、リンクが結果の受け渡しに相当する。静的データフローはリンクは一度しか用いることができずループを書いたり、グラフの一部を共用したり (関数に相当) することができない。動的データフローではリンク上を流れるトークンにタグをつけることでこれを解決する。計算の順序はデータの到着順序にのみ束縛される。

リダクション型: 式のグラフを (メモリ内に) 構築し、ゴールに相当するノードから順にたどって行き、簡略化可能な部分グラフを簡略化することによって計算を実行する。計算の順序は式が簡略化可能かのみ束縛される。

データフロー型はデータ主導、リダクション型は命令主導、ノイマン型は両方を同期させる (クロック主導)、ということか。

3.3 (単一の計算機という視点での) 限界

ノイマン型の限界:

- 並列化がむずかしい (副作用中心で計算を行うため)
 - メモリアクセスの排他制御
- メモリバスがボトルネックになる
- 命令とデータが同時に順序良く CPU に到着する必要がある
 - クロックが必要

データフローマシンの限界:

- 並列度の爆発
- 並列度/必要なメモリ量が事前に計算できない
- 計算に要する時間と、計算の準備に要する時間の比が悪い

リダクションマシンの限界:

- 実装がいちじるしく難しい (天野先生の講義資料より)

オブジェクト指向モデルの限界?:

- そもそもオブジェクト指向モデルに基づく計算とは何か、まじめに検討されているかどうか不明
- 「オブジェクト」の動作の表現が一つの計算モデルを構成しうるとすると、オブジェクト指向自身は計算モデルではない!?
 - ノイマン型プログラムの構造化手法 (プログラム分割+メッセージ交換) として発展したのは間違いではないのかも (副作用の局所化)

4 僕らはどこにいるのか: the Real Space Internet Blues

もういちどまわりをみわたしてみよう。インターネットとかユビキタス環境とかセンサネットワークとかオーバーレイとかがまわりに溢れている。一方、それを「うまく使う」方法は、個々の問題領域で毎回発明する必要がある。アーキテクチャの視点で言えば、まだ専用計算機の時代なのではないか？

4.1 現在の道具の整理

インターネット 省略

ユビキタス環境 とりあえず ID がつく (u-Code by 坂村, EPC)。ID と場所 (タグリーダ) の関連を管理する枠組みぐらいは共有化できるだろうか？

センサネットワーク 定義が不明だが、センサー (およびアクチュエーター) をネットワーク経由で制御可能であることを意味する？

オーバーレイ Rendezvous, Location, and Routing. 意味の変換 (抽象化)。スケーラビリティ。

4.2 計算の動機

我々が手持ちの道具を上手く使いたい、その動機を整理する。

データフロー型: データは貴重なリソースで、必要なときにあるとは限らない。あったらすぐ教えて。

リダクション型: データは現在ある中から使えればいいや。それよりも欲しいと思ったらすぐあるものから結果をよこして。

ノイマン型+オブジェクト指向: 勝手に作った個々の (単独の) プログラムの入出力を接続する。なにをどのように構成するかはそれぞれ詳細に設計する。(「Web2.0 でマッシュアップ」方式)

The Real Space Internet で必要になるのはどれか? Live E!の問題では個人的にはデータフロー型の立場に立っているが、人によっては (例えば気象解析屋) リダクション型に近い立場なひともいる。

図 1 的なものを見たときに、真ん中の Overlay の構成がデータフローで、下側の DB(っぽい) 連合の先にリダクション型が存在する? (右側の Attribute Table はデータフローマシンで言う連想記憶に相当するか)

もちろん、今実現可能、かつ責任分界点が明確である、という意味で、現在主流なのはノイマン型計算機の入出力をメッセージ交換によって制御するものである。ただ、これは単一の計算機へのプログラムを構造化するための計算論であり、インターネット (など) の上にプログラムを展開するための計算論ではない。また、計算可能かどうか計算前には不明かもしれない。命令とデータ (メッセージ) が必要なときに必要な場所にある保証はない。

5 オーバーレイと計算論: open なまとめ

今までの議論をもとに、以下の問題を掘り下げなさい (各 10 点)。

5.1 オーバーレイ的データフロー

インターネット的データフローモデルについて、計算モデルとそれを実現するアーキテクチャのどちらかの立場から、考えられる問題点や利点を、既存のノイマン型 (+オブジェクト指向) アーキテクチャと比較しながら述べよ (10 点)。

5.2 オーバーレイ的リダクション

インターネット的リダクションモデルについて、計算モデルとアーキテクチャのどちらかの立場から、考えられる問題点や利点を、既存のノイマン型 (+オブジェクト指向) アーキテクチャと比較しながら述べよ (10 点)。