

新機能マシン N F M

---メモリベース A I アーキテクチャ---

阿江 忠、酒居 敬一、豊崎 剛

広島大学工学部

メモリベース推論はモデル表現が陽でないリアルワールド向きであるが、そのインプリメントに通常の超並列マシンを用いることはコストパフォーマンスの点で問題がある。メモリベース推論はメモリベースアーキテクチャで実現するのが適当であり、ニューラルネットを能動メモリとして採用することができる。一方、メモリベース推論のみでトータルシステムを構築するのは困難であり、通常の A I システムとの融合も必要となる。新機能マシン N F M は、知識の獲得、表現をメモリベースで行ない、知識の進化を協調動作により実現するシステムであるが、メモリベースのオブジェクトとプログラムベースのオブジェクトの協調動作を利用するのも特徴の一つである。

NEW FUNCTIONAL MACHINE NFM

---- Memory-Based AI Architecture ----

Tadashi Ae, Keiichi Sakai, Tsuyoshi Toyosaki

Faculty of Engineering, Hiroshima University

The conventional massively parallel computers are too huge to implement the memory-based reasoning that is appropriate to realize the real world. The memory-based architecture fits the memory-based reasoning, where the neural networks play an role of active memory. The total system, however, requires also the conventional AI technique, and therefore, it must be realized to combine the memory-based reasoning with the conventional reasoning. NFM, a new functional machine, is proposed for this purpose and its outline is described.

1. まえがき

小文では、真に非フォンノイマン型といえるアーキテクチャは、メモリベースアーキテクチャが望ましいという観点に立ち、メモリベースAIアーキテクチャの一つとして、設計ならびに構築中の新機能マシンNFMを紹介したい。このメモリベースアーキテクチャは、ソフトウェア面からは、人工生命やリアルワールド環境で成長するシステムなど新しい人工知能分野への応用向きであると同時に、素子技術としては、光技術、量子化機能素子への期待が大きいアーキテクチャである。

2. プロセッサベースアーキテクチャとその限界

アーキテクチャは

概念レベル（論理レベルを含む）

物理レベル

の2つに分けることができる。

概念レベルにおいて、プログラムとデータの対を対象に物理レベルの設計を考えると、2つのアーキテクチャ、すなわち、プログラム依存アーキテクチャとデータ依存アーキテクチャが存在する。しかし、どちらもアーキテクチャをプロセッサ設計の問題に帰結しているので、このようなアーキテクチャは、プロセッサベースアーキテクチャと呼ぶのが適当である。フォンノイマン型コンピュータの定義は人によって異なるところがあるが、厳密に考えれば、次のようになろう。

[定義]フォンノイマン型コンピュータ

フォンノイマン型コンピュータとはプロセッサベースアーキテクチャにより構成されたコンピュータである。

この定義を用いると、プロセッサベースアーキテクチャならば、どんなアーキテクチャもすべて（程度に差こそあれ）フォンノイマン型の域を出ていないことになる。

むろん、フォンノイマン型コンピュータの威力は偉大で、永遠に不滅である。それは、実行すべき対象であるプログラムが正しく陽に与えられるならば、プロセッサベースアーキテクチャが極めて自然なアーキテクチャであるからである。

プロセッサベースアーキテクチャにおいて、単体でのスピードの限界を打ち破るために、

1) GaAs 素子や Superconducting 素子など高速素子の開発

2) マルチプロセッサによる並列プロセッサ化

という2つの流れのあることはよく知られている。最近では、2) が花盛りで「超並列プロセッサの開発」というテーマもやや陳腐化してきている。超並列プロセッサ実現の隘路はプロセッサ間のデータのやりとり（通信）で、これを解決するのは、

高速クロスバースイッチ または マルチポート共有メモリの開発につくる。

筆者らは、マルチポート共有メモリを用いるアーキテクチャには古くから固執している。マルチプロセッサによる並列プロセッサの開発は結構古くから手掛けてきたが、資金的にも人的にも圧倒的に恵まれた組織が参入してきた時点で方針を変更した。むろん、それらの一部は商用的にも成功しているし、今後、着実な進歩を遂げることは間違いない。

現時点では、アーキテクチャとしての問題点よりむしろ、並列プログラムをどう書いて、コンパイルするかというソフト面に問題点は移行している。前者の「並列プログラムをどう書くか？」という問題は大きな問題である。10⁴程度までの並列ならともかく10⁶とか10⁸という超並列になったときは、プログラムのもとになる計算式やモデルの是非まで再

考しなければならない。これを「超並列計算モデルの問題」と呼んでおく。

いずれにしても、マルチプロセッサや並列プロセッサのアーキテクチャもプロセッサベースアーキテクチャである。したがって、やはりフォンノイマン型コンピュータである。筆者の一人（阿江）の比較的最近のサーベイ論文「非フォンノイマン型チップアーキテクチャ」でも、マルチプロセッサ自体には触れていないのも、この理由からである。

3. 低粒度並列性の実現

そもそも、コンピュータ内部の並列化は、

- 1) プロセッサレベル
- 2) 演算器レベル
- 3) ゲート素子レベル

に分けられる。プロセッサベースアーキテクチャの並列化は、ふつう、1)と2)を対象とする。2.で述べたように、並列プロセッサ実現の隘路は「プロセッサ間通信」であるが、

通信時間／プロセス実行時間

をオーバーヘッドのパラメータとすると、分母のプロセス実行時間が長いほど、通信時間への要求は緩くなる。その意味では、プロセッサを並列化するマルチプロセッサは妥当な方法といえる。そして、2)、3)と演算単位（「粒度」という）を小さくするほど、通信への要求がきつくなり、当然、実現が困難になる。データフローコンピュータは命令レベルで並列化するのが自然であるが、プロセッサ間通信への要求は極めて厳しいから、その実現コストは非常に高いものとなり実用化は容易でない。

しかし、「ものは考えよう」であり、粒度が小さくても、通信の行き先が固定できるならば、実現は難しくない。事実、2)の演算器レベルでも、シストリックアーキテクチャは隣接する演算器へのみデータを転送することで実現を容易にしているので、超並列向きアーキテクチャである。

もう一つ粒度を下げると、3)のゲート素子レベルの並列性である。「ゲート素子」は、コンピュータアーキテクト向けの表現で、これは単に「素子」と言い換えてもよい。そうすると、自然にアナログ素子的なものも含まれることになる。プロセッサを論理レベルまででとらえている人からみれば、狂気の沙汰かも知れない。このゲート素子レベルで相互結合すると安定状態をつくり、メモリとなる。メモリベースアーキテクチャの構成基本単位は（アナログ素子を含む）このレベルにある。メモリベースアーキテクチャは素子レベルの並列性を使うため、自然に超並列なシステムが実現できる。

4. メモリベースAIアーキテクチャによる新機能マシンNFM

前置きが長くなったが、180度発想を転換してみよう。概念レベルのプログラムとデータを、物理レベルにおいて、プロセッサへ写像するのではなくて、メモリへ写像するのである。このとき、メモリの機能がアップし、メモリがプロセッサの役割をも（ある程度）する必要が生じる。機能メモリあるいは能動メモリといわれるメモリがこれに相当する。このように、メモリに機能をもたせ、能動的にしたアーキテクチャをメモリベースアーキテクチャと呼ぶ。新機能マシンNFMはメモリベースアーキテクチャにより構成される。

概念レベルにおいては、コンピュータへ要請する処理の表現方法が

1) 論理的な表現

2) それ以外

に分けられる。1) の「論理的な表現」とは、コンピューティングのモデルが陽であり、なんらかのプログラム記述が可能となることを意味している。したがって、A I (人工知能) システムであっても、ルールやフレームなどの知識表現が得られた時点で、プログラム記述はできることになる。その意味で、従来のプロセッサ設計では、「概念レベル」とは言わず、「論理レベル」と呼んでいる。

ところで、概念レベル＝論理レベル、とした1) のタイプのメモリベースアーキテクチャも存在する(例えば、FMPP)。このタイプの特徴は、メモリベースではあるが、「プログラムは論理的に記述される」という点にある。拡大解釈すれば、セルラーオートマトンをマシン実現したもの(前述のシストリックアーキテクチャもその一例)もこのタイプに含まれる。メモリの単位をセルと呼び、セルに処理機能を付加したと考えられるからである。さらに言えば、セルラーオートマトンを考案したのもフォンノイマンなのである。

プロセッサベースアーキテクチャから逃れて、さらに、論理的にプログラムを記述するというタイプも除外することで、やっと「ノイマンの亡霊」から逃れることができる。コンピュータが真に非フォンノイマン型であるためには、概念レベル表現は論理的に陽ではないほうが望ましい。2) それ以外、すなわち、「論理的ではない表現」という項目を設けた理由はここにある。

論理的には陽でないシステムとしては、真っ先にニューラルネットが挙げられよう(いわゆるA I システムと異なり、知識獲得ののちの知識表現が陽でなくとも実用になるシステムである)。そして、ニューロンを相互に接続したニューラルネットは、通常の論理素子の代わりにニューロンという素子を基本単位とするシステムになっている。

コンピュータアーキテクチャという観点からみたニューラルネットの特徴は、次のようにまとめられる。

1) 2値論理素子(デジタル素子)の代わりに、ニューロンという実数値をとる素子(アナログ素子)を基本単位としている。

2) 素子レベルの並列性をもつ。

1) は学習を容易にするほか、離散計画問題を実数補間することでハードな問題を近似的に解くのに役立つし、2) は高速化に多大な貢献をする。とりわけ、組合せ最適化問題の近似解法を提供するHopfield Netは、当初、強烈なインパクトを与えた。この場合、ユーザの与えた問題がうまくマッチすれば、ニューラルネットだけで問題は解けることになる。しかも、この2) の特徴こそ、ニューラルネットがメモリベースシステムの一例であることを示している。ニューラルネットを連想メモリあるいはニューラルメモリと呼ぶのはそのためであり、機能メモリの一つに位置づけられる。

しかし、多様な用途を対象としたとき、柔軟性に欠けるという欠点をもつ。最大の理由是非線形関数のコントロールが難しいという点にある。この欠点を補うために、外からコントロールするのが自然な拡張であろう。もっとも、コントロールが強力になると、コントロールのアルゴリズムだけで問題を解くことと同じことになり、ニューラルネットの効果は消滅してしまう。つまり、特定のニューラルネットモデルを基本単位として採用するのは妥当ではない。

いずれにせよ、ニューラルネットは単体のままではA I (人工知能) システムとしては能力不足である。

[ニューラルネットの能力]

通常のニューラルネットにはA I システムで必要な知識獲得の機能はあるが、

知識表現の機能はない。

これは、ニューラルネットは数値のレベルでの処理を基本とするためであり、当然でもある。例えば、(身長、体重)の組の数値を入力し、判定結果を教えて学習を行なわせると、学習終了後には、任意の(身長、体重)を入力すると「痩せ過ぎ」「ふつう」「肥え過ぎ」の判定結果を出力するようになる。しかし、ニューラルネットでは記号による知識表現は得られない。「肥満の人は、身長に比べ、体重が多い」あるいは「(身長-100) × 0.9を境に判定するプログラム」のような知識表現が得られるとニューラルネットもAIシステムと言えるようになる。

このような能力をもつメモリシステムは、次のような2段階で構成するのが適当であろう。

知識獲得 (Step 1):

知識獲得メモリ---データを整理して知識獲得を実現

知識表現 (Step 2):

知識表現メモリ---整理されたデータをもとに知識表現を実現

それぞれのステップで用いられるメモリは、当然、能動メモリである。それぞれのメモリへの要請については十分な検討が必要である。何故なら、能動メモリで実現可能なものは制約が多いから、それを考慮しないと概念レベルの設計が能動メモリで実現不可能となるからである。

このようにメモリベースアーキテクチャの振舞いも、通常のAIアーキテクチャと同様、知識獲得と知識表現の2ステップに分けられている。しかし、いずれのステップもメモリ(能動メモリ)がその実行を行なうところに特徴がある。知識獲得メモリをAメモリ、知識表現メモリをRメモリと呼ぶ。それぞれのメモリへ要請される機能をまとめると次のようになる。ただし、実現可能性は?であるので、理想という注釈を括弧付けしておく。

[Aメモリへの要請(理想)]

位相をもつこと。つまり、データの集合はランダムな集まりではなく、位相空間上の点の集合として存在すること。

[Rメモリへの要請(理想)]

データの集合がつくる位相空間において、関数(またはルール)と等価な「クラス(カテゴリ)の間の関係」が定まること。

これら能動メモリへの要請の説明と現実的な対応を以下で述べる。

5. 知識獲得のための能動メモリ --- Aメモリ ---

相互結合によるフィードバックをもち閉じた系をなす能動メモリは、(一般性をもたせるのはここ当分困難なので)ここではあきらめることにする。そこで、メモリに若干の機能(Function)を付加して「知識獲得」に役立たせるという方針をとることにしよう。機能付加の一例を示そう。通常のメモリではランダムにデータが格納されるが、ソーティング機能を付加すると、整列した順にデータを読みだせる機能メモリが出来上がる。(実数データをソーティングすると、結果のデータ空間は一次元実数空間の部分空間になっている。)

ソーティングは頻繁に用いられる重要な機能ではあるが、それ自体並列処理の対象となるくらい手間のかかる処理である(ニューラルネットが理想的に構成できるのであれば簡

単にできる)。さらに、現実に使われるデータは、一般的には多次元であるため半順序集合を扱う必要があるし、また、数値でないデータに対しては前処理も必要となる。これらの事情を考えると、前述の[Aメモリへの要請(理想)]をそのまま実現するのは現実的ではない。

一方、プロセッサ側からみれば「ソートはサーチを容易にする前処理」と解することもできる。もっとも、ここでいう「サーチ」とはちょうどピッタリのものがないときは、一番近いものを捜すという意味をもつものとする。このとき、サーチされるデータ空間は仮想的には(探索点からみて)整列されていると考えたとしても、実際は整列されていなくとも処理にはさしつかえない。そして、並列サーチを使うとすると処理時間は大幅に短縮できるので、コンピュータをリアルワールドで用いるときには、極めて重要となる。

このように、きちんと整ったデータ空間をつくる機能を要請する代わりに、並列サーチ機能の実現を要請するほうが現実的である。そこで、Aメモリへの要請を理論的なものから、次のような現実的なものへ変更する。

[Aメモリへの要請(現実)]

データ集合の中から一番希望に近いデータを高速に見つけること。

実は、この機能はメモリベース推論(Memory-Based Reasoning、略してMBR)において要請されているものである。現在の入力に対してもっとも適切なレスポンスをするためには、過去のすべてのデータ(入力とレスポンスの組)の中から、もっとも近いもの捜せばよいという考えかたに基づいている。例えば、新しい入力データ(身長、体重)に対する判定は格納されているデータすべての中から、もっとも似通ったものを捜し、その結果を使う。このメモリベース推論は事例ベース推論(Case-Based Reasoning、略してCBR)の一種として知られている。

以上のように要請したAメモリは、AIシステムにおける知識獲得に相当する部分を、メモリベース推論的な意味で実行している。

6. 知識表現のための能動メモリ --- Rメモリ ---

Aメモリは、量的な問題を別にすれば、原理は簡単である。このAメモリへの要請と同じ原理のメモリベース推論MBRは、すでにプロセッサベースアーキテクチャではいろいろな使われ方をしている。MBR自体は概念レベル表現なのであるが、これまでのインプリメンテーションはいずれもプロセッサベースアーキテクチャに関するもののみである。これに対し、筆者は1992年にメモリでのMBR的なものの実現を開始した。そこには、ここでいうAメモリ相当なものも含まれている。

Aメモリすなわち知識獲得だけでは、半分の機能しかもたない。というのは、AメモリベースのAIシステムはこれまでのプロセッサベースシステム(フォンノイマン型)の不得手な分野を得意とするけれども、このAIシステムは、また、プロセッサベースシステムの協力を必要としている。プロセッサベースシステムとのやりとりのために、知識獲得のあとの知識表現が不可欠である。

そこで登場する知識表現メモリすなわちRメモリはプログラムと同等の働きが要請される。ただし、一般のプログラムのような記号列の表現を得るのではなく、等価な表現が得られれば良いとしよう。Rメモリへの要請も「理想」から「現実」への妥協は必要であり、具体的には、次のようなものが適当であろう。

[Rメモリへの要請(現実)]

データの集合がつくる位相空間において、クラス(カテゴリ)の代表を得るこ

と。

このためのアルゴリズムとしては、厳密には統計的処理が必要であるため、時間がかかるし、能動メモリの枠外になる。近似的には、「乱数によるサンプル抽出および自己組織化」が適当な方法と考えられ、能動メモリへの付加も極くわずかですむ。近似の精度とアルゴリズムの実行時間は逆比例するが、精度の指定はユーザが自由に選べるようにできる。

7. 知識オブジェクト進化システム

Rメモリが知識表現能力をもつとしても、単独では、能力に限界がある。そのため、複数のRメモリとおしの協調動作も必要である。メモリがそれに作用するプログラム（手続き）を内包するようになると、オブジェクト指向モデルにおけるオブジェクト相当となる。Rメモリを一つのオブジェクトと見たてると、Rオブジェクトと呼ぶことができる。Rオブジェクト間の相互作用による協調動作は知識オブジェクトを進化させるが、知識オブジェクト進化のプロセスは、Rメモリの集まりばかりでなく、Rメモリと通常のAIシステムにおける知識表現（これもオブジェクト化して、Pオブジェクトと呼ぶ）との協調も重要になる。

前述の知識表現（Step 2）は「プログラムとしての表現が求められること」を意味したが、オブジェクトモデルの世界では

知識表現（Step 2）：Rオブジェクト生成

という表現になる。Rメモリがその役割をすることに変わりはない。Pオブジェクトは従来の方法で求められるものとする。そうすると、

知識表現の高度化（Step 3）：

オブジェクト相互作用---知識オブジェクト進化システムの実現
という新しい段階の導入としてとらえることができる。

オブジェクト進化のパターンは次の4通りになる

- 1) PオブジェクトーPオブジェクト間：従来と同じ。
- 2) RオブジェクトーRオブジェクト間：同値類上での推論。
- 3) PオブジェクトーRオブジェクト間：Pが主体でRがその補助をする。
- 4) RオブジェクトーPオブジェクト間：Rが主体でPがその補助をする。

新機能マシンNFMの設計は、能動メモリからなるオブジェクト進化システムの詳細な概念レベルと物理レベルの設計を述べなければならないが、紙面の都合上省略する。

8. 新しいデバイス技術への期待

--- 光インターコネクションと量子化機能素子 ---

光インターコネクションは、光電子デバイスのなかで有望視されているが、ここでは「何故光を用いる必要があるのか？」という点のみ要約して述べるにとどめる。

光インターコネクションの特徴はバンド幅が広いことであるが、そのバンド幅の広さは「サーチ」にこそ特徴が発揮される。メモリベースアーキテクチャでは、

- 1) AおよびRメモリにおけるサーチ

これはバスを用いるサーチがOEICのようなチップ内で効果的であることを意味している。

- 2) 知識オブジェクト間の相互作用におけるサーチ

これも、データ量が比較的小さいときは、サーチ専用OEICが対応するが、データ量が大きくなるとデータのやりとりはプロセッサが補助的に介入するようになる。このプロセッサは通信専用であるが、光インターコネクションはMCM（Multi-Chip Module）の

技術として重要な役割をもつ。

量子化機能素子への期待は「メモリの大容量化」という言葉で要約される。メモリベース推論でも大容量のデータ（事例）がリアルワールドから入力されたところから、処理は始められる。メモリベースシステムは、データ量の多さを避けるのではなく、反対に歓迎する。高度に複合化してモデルの数式化の困難なシステムは超並列計算モデルをつくること自体が難しい。これが「超並列計算モデルの問題」である。しかし、計算モデルはなくともデータは山ほどあり、大容量メモリには容易に貯えられる。メモリデバイスを量子機能素子により実現できれば、いくらでもデータの格納は可能になる。その上でメモリベース推論を行なえば、「超並列計算モデルの問題」は回避できることになる。なお、メモリベースシステムでは、（ある程度の誤差を許容する処理を前提にしているから）メモリの内容がビット単位で厳密に正確でなくてもよい。このことは量子化機能素子の動作と相性はよく、量子化機能素子の開発方向を示唆しているともいえよう。

9. むすび

メモリベースAIアーキテクチャをもつ新機能マシンNFMの概要を述べた。メモリベースアーキテクチャは、処理のモデルが陽でない問題の解を求めるのに適しておりリアルワールド向きである。もともと、リアルワールドの問題は超並列であるが、計算モデルを正確に求めるのが困難であることが多い。したがって、超並列コンピュータをプロセッサベースアーキテクチャで構築しても、「超並列計算モデルの問題」を避けるには、結局、「概念レベルではメモリベースの表現を用いる」ことになる。それなら、「物理レベルでもメモリベースで実現すればよいのではないか」というのが、ここでの主旨である。

メモリベースアーキテクチャのつらいところは、概念レベルではそうであっても、物理レベルではついマルチプロセッサに走ってしまうところにある（CM-1がその典型的な例であろう）。いくらメモリベースと言っても、メモリに付随する処理がデータに依存してしまうとデータごとに処理を書くことになり、結局プロセッサの手を借りることになる。

したがって、メモリベースを貫徹するには、データに依存しない処理のみをメモリに付随させて、Aメモリ、Rメモリ、オブジェクト進化システムをつくる必要がある。そのためには、概念レベルをきっちりさせてから物理レベルの設計を始めなければならない。筆者の提案するNFMの構成でも、メモリベースアーキテクチャと呼べるのは、現時点では、AメモリとRメモリであり、残りはマルチプロセッサの援助が必要である。しかし、根気強く足元から一步一步メモリベースアーキテクチャを進展させることが肝要であり、現存するアーキテクチャとは別の新世界を切り開くための同志を募っているのが現状である。物理レベルでメモリベースといえるものは、これまでのところ、いずれも、国産(?)である。半導体の生産のうち、メモリの分野で群を抜く我国の場合、プロセッサのことはさておき、メモリベースアーキテクチャに活路を開くのも一案ではないかと考えている。

注：紙面の都合上、図および文献を省略した。同じような主旨の文は著者の一人（阿江）が雑誌（例えば、電子工業月報、1994年4月号）などに発表しており、それらの参照ということにしたい。