4D−12

日本語情報処理用の計算機システム

― 文書処理に適したデータ構造とメモリ構成 ―

石松 謙治，寺澤 登紀江，大須賀 勝美，黒川 一夫

東京理科大学

1．はじめに

現在のコンピュータは欧米語圏で開発され、数値計算を行うことを目的としている。そのため、日本語の処理に適しているとは言い難い。そこで本稿では、日本語の特徴を生かした、新しいアーキテクチャを持つ日本語処理専用システムの開発を目標とし、日本語文書の処理に適した文字列データとしての取り扱い方法とそのための新しいメモリ・アーキテクチャについて検討を行う。

2．日本語文章の取り扱い

2．1 一般的な日本語処理

一般的なテキストデータでは、文章を単なる文字の並びとしてしか扱っていない。そのため、文章を１文字づつ処理を行っている。しかし、日本語の文章はいろいろな長さの熟語によって構成されているとも考えられる。そこで、文章を熟語ごとに区切って処理を行う方が効率的な処理が期待できる。

しかし、コンピュータが数値演算を行うことを目的としているため、これまでの日本語処理では処理できるデータ長がメインプロセッサのもつデータバス幅に規制されていた。

また、文章の処理は検索、並べ換えといった単純な作業が大部分を占めるが、そのデータ量が大量であるため、そのような作業を高速に処理する能力が要求される。

2．2 熟語情報、結合情報

日本語文章データをその文字種類により熟語単位に分解して、熟語情報と結合情報という２つの情報に分けて取り扱うことを考える。熟語情報は文章データ内で使われている熟語を集めて整理したものであり、結合情報は熟語情報の文章データ中での並び方を表したものである。

一つの文書中には、しばしば同じ熟語が何度も現れる。この傾向は専門的な文書であるほど顕著である。そこで、重複する熟語を熟語情報として一つにまとめることにより検索などの効率を上げることができる。

このように熟語情報として文章データを取り扱うことにより、シリアルであった文章データの流れをパラレルに処理することができる。

2．3 日本語コード

一般的な日本語処理では、漢字もアルファベットもみな同等に扱われており、漢字の表意文字であるという特徴を生かしていない。

そこで本研究では、従来の文字コードとは異なり、漢字の部首情報を漢字コードに利用することにする。平仮名、片仮名、アルファベット、数字、記号といった文字は特殊な部首に属しているものとして取り扱う。

3．メモリ・アーキテクチャ

文書データに関して熟語ごとに任意の長さで処理を行うため、従来のメモリとは異なるアーキテクチャを持ったメモリが必要である。そこで、文書データはプログラムとは分離して、文書データ専用のメモリを用意する。

このメモリは、固定されていない幅のデータを一括して処理するために、複数のメモリプロセッサとそのおのおのに付属する複数のサブメモリ、加えてメモリプロセッサを統合、管理するメモリ・マネージャによって構成されている。

図１に本研究で考えている日本語処理システムのメモリ・アーキテクチャを示す。

図２にメモリプロセッサの仕様を示す。

１個のメモリプロセッサに割り当てられるデータ幅は１６ビットで、文字１文字分を表現するのに必要な幅である。

サブメモリには高速メモリを使用し、サブメモリの容量を越える量のデータを処理するために、各メモリプロセッサにはそれぞれ大容量記憶装置がつながっており、サブメモリと併せてメモリプロセッサの仮想メモリとして動作する。

各メモリプロセッサは、メモリ・マネージャによって管理されており、それぞれ独立しても、連結しても動作することができる。

Computer System for Japanese Language Processing
― Data structure and Memory Architecture ―
Kenji ISHIMATSU, Tokie TERASAWA, Katsumi OSUGA, Kazuo KUROKAWA
Science University of Tokyo

4．検索への応用

日本語処理の作業の重要な部分を占めるものとして検索作業がある。ここでは、このシステムで検索作業を行った場合について検討する。

文書データの検索を行う場合、①結合情報を検索する場合と、②熟語情報を検索する場合とがある。

①結合情報の検索

メインプロセッサの指示を受けて、各メモリプロセッサは結合情報を外部メモリから一斉に読み出して、サブメモリに書き込む。

各メモリプロセッサは並行して、送られてきたデータと比較しながらサブメモリ内のデータを検索していく。

検索に成功したならば、１ビットの検索成功信号をメモリ・マネージャに返す。

②熟語情報の検索

メモリ・マネージャが熟語長に合わせてメモリプロセッサを連結させてメモリプロセッサ集団を形成させる。

メインプロセッサの指示を受けて、各メモリプロセッサ集団は熟語情報を外部メモリから一斉に読み出して、サブメモリに書き込む。

各メモリプロセッサ集団は並行して、送られてきたデータと比較しながらサブメモリ内のデータを検索していく。

検索に成功したならば、１ビットの検索成功信号をメモリ・マネージャに返す。

熟語情報を検索するとき、その種類の熟語情報のみをサブメモリに呼び出して調べることにより、検索量の大幅な削減を図ることができる。

5．まとめ

以上の結果より、次のようなことがいえる。

・文字列データをその熟語種類別、熟語長別に分けておくことにより、検索範囲を大幅に削減することができる。
・文字列データを高速に取り扱う作業領域としては、並列的に動作できるメモリが適している。
・複数のメモリプロセッサを連動させることにより、任意長のデータを一括して取り扱う事ができる。
・メモリプロセッサを並列に動かすことにより、作業を高速化することができる。
・文書データの入出力、検索、並び変えの作業をメモリプロセッサに行わせることにより、メインプロセッサの負担を大幅に軽減できる。

ここで取り扱っている熟語情報と結合情報を用いて検索シミュレーションを行った結果、大きな文章中からの検索も目的とする文章を簡単に見つけ出せる事が確認されている。

さらに、このシステムを発展させて文章の解析、理解、さらには人工知能などへの利用も考えられ、一般的なシステムよりも効率的な処理が期待される。

＜参考文献＞

[1]大須賀勝美，黒川一夫：
“日本語処理用の計算機システム”
第42回　情報処理学会論文集(6)，
p106-109，(1991.3)

[2]高橋恒介：
“文字照合処理への応用”
情報処理学会論文誌，Vol.32，No.12，
p1268-1275，(1991.12)

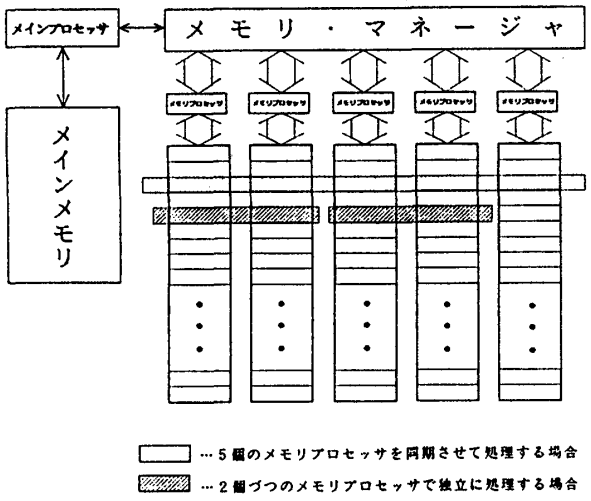

図１　並列メモリ

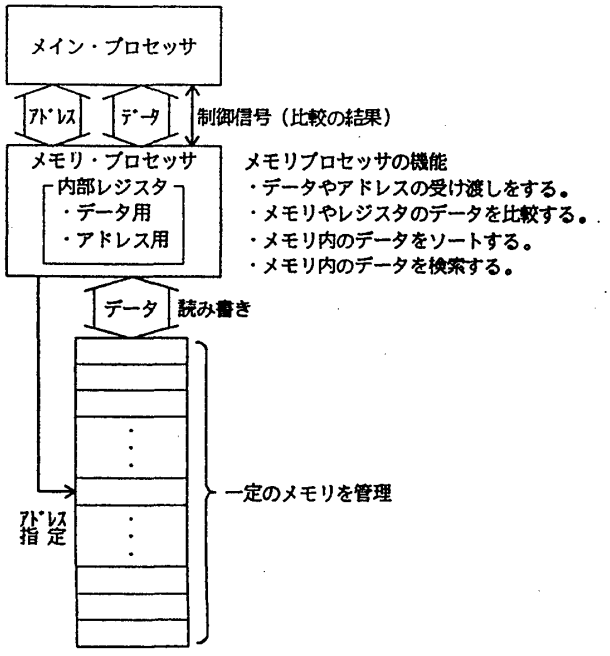

図２　メモリプロセッサ