

市販並列パソコンを用いた逐次プログラムの並列化の練習

山崎 信孝¹・本田 数博²

¹ 広島大学大学院数理分子生命理学専攻博士後期課程
(神奈川工科大学大学院応用化学専攻博士前期課程修了)

² 応用化学科

The practice of the parallelization of the serial program
using the marketing parallel personal computer

Nobutaka YAMAZAKI・Kazuhiro HONDA

Abstract

The parallel program was constructed using parallel processing board (mp98) of the personal computer correspondence which can easily handle parallel computer. The calculation speed was then measured by using parallelization of the primality test processing for showing the difficulty of the construction of the parallel program. As the result, the processing speed was drastically different by the method of the parallelization, i.e., the calculation time became shorter in the case of dynamic load dispersion.

Key Words: Parallelization, Personal Computer.

1. はじめに

近年、コンピュータシミュレーションによって化学現象を探る研究が盛んであり、このような研究にはプログラムの構築作業が不可欠である。一般に、プログラムの構築においてユニット(例えば物質・相・エネルギー等)とユニット間の相互作用の計算になるとユニット数の増加に伴い計算量は莫大(極端な例では階乗に比例)に増加してしまう。昨今のコンピュータの性能向上には目を見張るほどであるが、要求される計算量はそれをはるかに凌駕するほどである[1,2]。それゆえコンピュータの負荷軽減のための1つとして並列化が有力である。

並列化には同じ計算を同時に行なうことによって個々のプロセッサの処理を軽減する並列処理と各々のプロセッサに別々の処理を割り当てる分散処理がある。1個のプログラム(逐次処理)の書き換えにおいて並列処理は各々のプロセッサが同時に計算し、同期を取りながらまとめればよい。しかし、分散処理では其々のプロセッサに対し別のプログラムを割り当てているため、場合によってはあるプロセッサの結果を別のプロセッサが使用するというような独立で計算できない場合の処理をしなければならない。従ってプログラムは同

期及びデータの転送を考慮して構築しなければならないのでかなり困難になる。ところが、銀河内の星の運動の計算ような同時刻に独立で計算できる系はまれであり、大抵の場合は環境や生体内の循環反応のような逐次的なものの計算を軽減したいのが一般的であり並列化は分散処理にせざるを得ない。そのため容易な題材で並列プログラムを組む練習が必要である。残念ながら、いきなり化学を題材にしたサンプルプログラムの並列化は非常に困難であり分散処理の本質を把握し辛い。それゆえ、教材としてあえて数学的な題材である素数の判定処理を用いた。素数判定プログラムは自然数を素数の候補とし、それに対し2,3,5,...の素数で1個でも割り切れれば素数ではないものとした。そのプログラムの分散方法としては候補を順番にプロセッサに割り当てる方法(静的負荷分散(Static Load Dispersion)―即ち計算前にプロセッサに割り当てている)で行なうと、あるプロセッサは必ず2,3などの倍数になりその判定は早く終わってしまう。従ってそのプロセッサが早く終わる分、他のプロセッサの負荷が大きくなり全体の処理向上につながらない。それに対して、処理中に空いているプロセッサに分散する方法(動的負荷分散(Dynamic Load Dispersion))では理想的な並列化がで

きる。我々は安価に購入できる並列処理ボード mp98 [3]を用いることによって素数の判定において静的負荷分散と動的負荷分散の違いから計算速度の相違を感じさせることを目的とした。

2. 計算

計算は1億から1億1千までに存在する素数判定を行なった。計算の流れを Fig.1 に示す。まず逐次処理(Fig.1A)では、"preparation"で1億から1億1千までの判定に必要な素数を用意する。"The prime number candidate is made."は、1億から1ずつ順番に候補を立てる。次の条件"The maximum prime candidate ?"で1億1千を越えた時点で終了とする。"Primary Test"では候補に対し2,3,5...と素数で割り切れるかどうかを判定する。もし素数であれば記録

する。その後"The prime number candidate is made."に処理を飛ばす。並列処理は"Primary Test"の部分を実行化した(Fig.1 B)。変更後のフローチャート中の"Distributed Processing"で、静的負荷分散と動的負荷分散の2通り行なった。静的負荷分散は総プロセッサ数を P_n とするとプロセッサ $P(P|P = 0, 1, \dots, P_n - 1)$ に $100000000 + P + P_n \cdot i$ ($i \geq 0$ の整数)の候補を処理させた。一方、動的負荷分散は実行中にプロセッサの空き状況を常に監視し、空いているところのプロセッサに処理させた。

計算条件は静的負荷分散及び動的負荷分散に対して、プロセッサ数を1~4までの並列処理を行い"preparation"を除いた個々のプロセッサの処理時間を計測した。また、計算機仕様は Table 1 に示した。

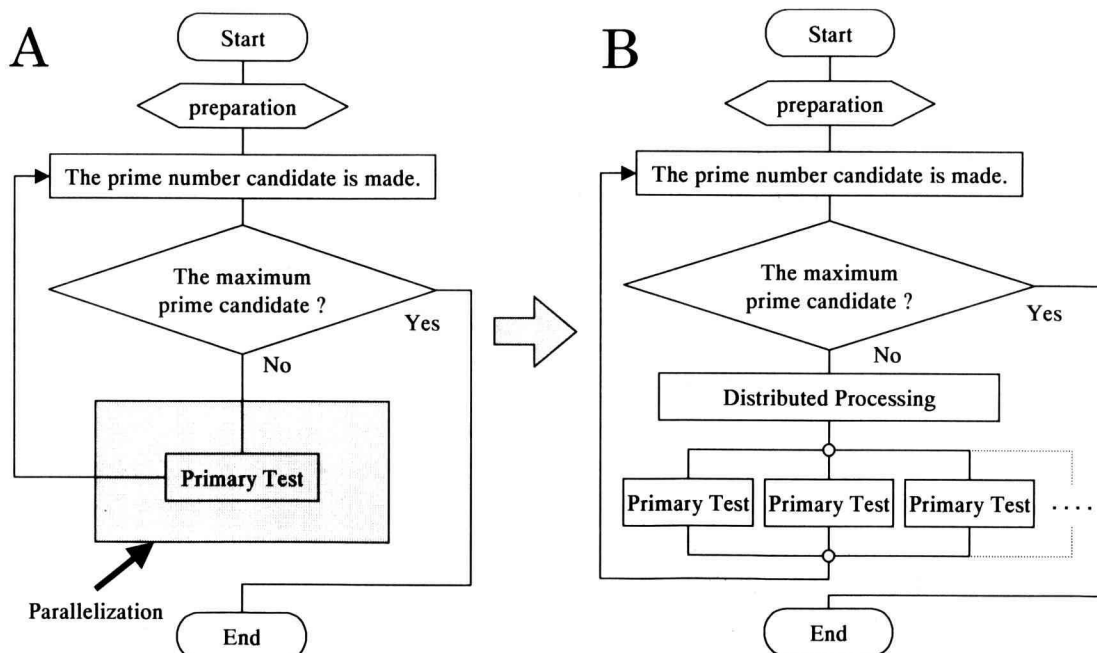


Fig. 1 The flowchart of sequential and parallelized operation

Table 1 The specification for Parallel Computer (mp98 board)

Processing Element	V50 (8MHz)
Number of Processing Elements	4
Connection Type	Bus Connection, Shared Memory Type
Shared Memory	256kB
Local Memory	256kB per processor
Host Computer	PC-9801 series (after VM), and PC-9821 series
used Address in Host Computer	DE000:DFFFF (static)
Operating System	MS-DOS on the host computer, but mp98 board has no.
Language	C language by LSIC-86 Compiler and Parallel Computation Library

Table 2 The Parallel effect in the primarity test processing

Method	P_n	Calculation Time / ms	P.E. Id 0 / ms	P.E. Id 1 / ms	P.E. Id 2 / ms	P.E. Id 3 / ms	The unit number effect	The parallel efficiency
Static Load Dispersion	1	7936	7936	—	—	—	1.00	100.0%
Static Load Dispersion	2	7876	62	7876	—	—	1.01	50.4%
Static Load Dispersion	3	4246	4246	3636	56	—	1.87	62.3%
Static Load Dispersion	4	4194	32	4194	31	3686	1.89	47.3%
Dynamic Load Dispersion	1	8041	8041	—	—	—	1.00	100.0%
Dynamic Load Dispersion	2	4034	4007	4034	—	—	1.99	99.7%
Dynamic Load Dispersion	3	2705	2675	2705	2662	—	2.97	99.1%
Dynamic Load Dispersion	4	2023	1996	2023	2008	2014	3.97	99.4%

※ : P.E. Id 0, P.E. Id 1, P.E. Id 2, and P.E. Id 3 show the processing time of the processor with each Id (0-3) for the calculation.

3. 結果

解は素数の最大は 100000969, 最小は 100000007, 素数の数は 54 個である。それぞれの条件で得られた計算時間を Table 2 にまとめた。計算時間 (Calculation Time) は一番時間を費やしたプロセッサの時間である。台数効果 (The unit number effect) はプロセッサが 1 個のときの計算時間をプロセッサ数 P_n 個の計算時間で割った値であり実質的に働いたプロセッサ数を示す。並列効率 (The parallel efficiency) は台数効果を P_n で割った値であり、計算時間上どれだけ並列化されているかを示す。また P_n の変化に対する計算時間の変化を Fig. 2 に示した。

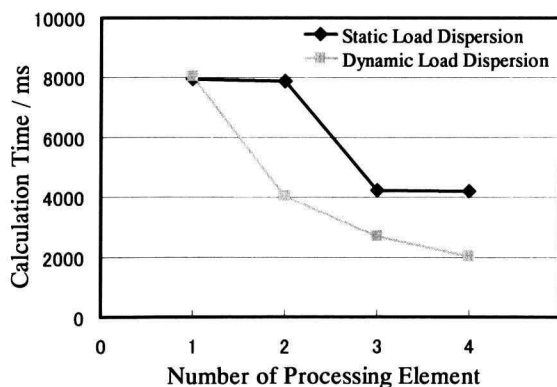


Fig. 2 The change in the calculation time with the increase in processor number

静的負荷分散では、 P_n を増加すると $P_n=2$ の時には並列化したときの計算時間の短縮にはつながらずプロセッサ 0 は殆ど働いていなかった。 $P_n=3$ においては時間短縮につながったが、プロセッサ 2 は殆ど働いていなかった。同様に $P_n=4$ においてもプロセッサ 0, 2 が殆ど処理を行っていないことがわかった。その原因は規則的に素数の候補を振り分けたため $P_n=2$ の時にはプロセッサ 0 は必ず 2 の倍数、 $P_n=3$ の時のプロセッサ 2 は必ず 3 の倍数、 $P_n=4$ ではプロセッサ 0, 2 が必ず 2 の倍数であるから判定時にはすぐ素数ではないと判断でき結果的に他のプロセッサに負荷が大きくなり全体の計算時間が大きくなったためである。一方、動的負荷分散では、 P_n を増加すると計算時間は反比例的に減少していた。これは理想的な並列化を示していた。

4. 結論

あえてハイスペックではない並列パソコンを用いることでむしろ、分散の違いによる計算時間の違いが顕著に表れた。並列化による処理の高速化は、いかにしてプロセッサを効率よく用いるかというソフトウェアの役割が非常に大きい。従ってユーザーがモデルの分散化に対する結果を計算速度として実感でき分散化の構築に再考を与えるものとして良いと思われた。

引用文献

- [1] 山本 芳嗣・久保 幹雄著 巡回セールスマン問題への招待 朝倉書店 1997/2
- [2] 笠井 琢美・戸田 誠之助著 計算の理論(情報数学講座) 共立出版 1993/6
- [3] 小畑 正貴著 つくる並列コンピュータ 東京電機大学 1995/3/30