輪講資料

津田 健

平成９年５月２９日

1　並列コンピュータ

1.1　はじめに

現在, コンピュータは非常に高速化されている。しかし、このような高速なコンピュータをもってしても, 長時間の計算時間を要する計算が多数存在する。

そのような計算を高速に処理させるためには, 並列に処理処理させるしかない。これから、並列方式の諸方式及びそれに関する特徴を述べる。

2　並列処理の形態

並列処理に関して, いろいろな分類法をのべる。まず, 処理の仕方について

- 逐次処理
- パイプライン処理
- パラレル処理

の3つに分類される.

パイプライン処理は, 処理を複数の段階に分けて, 次々と結果を渡す形で処理を行う.

パラレル処理は, 処理を互いに独立な複数の段階に分割して, 処理を行う.

3　並列処理の単位の大きさ

並列処理の単位の大きさ (粒度) によって,

- 細粒度
- 中粒度
- 粗粒度

の3つに分類される.

粒度の目安として, 命令を単位とする細粒度, ループを単位とする中粒度, ブロック, サブルーチンを単位とする粗粒度などに分かれる.

細粒度 ~ 中粒度並列処理の単位とする命令の系列は, スレッドと呼ばれる.

4　並列コンピュータの諸方式

現在の並列計算機においては、

- 制御駆動型
- データ駆動型
- 要求駆動型

の3つに分類される.

制御駆動型とは, 命令流が全体を制御しており, フォンノイマン型ともよばれる。

それに対して, データ駆動型は, 各演算で必要とする入力オペランドがそろったときにその演算を実行し, 要求駆動型は, 値が必要となった時点で計算を起動する. つまり, 命令流以外のものが全体を制御しており、これらは非ノイマン型とも呼ばれる.

これから、Flynn の分類に従って, 制御駆動型の並列計算機について, 述べる.

Flynn は命令 (Instruction) とデータ (Data) の流れが単一 (Single) か複数 (Multiple) かによって以下の４つの方式に分類した.

SISD (Single Instruction Single Data Stream)
単一命令単一データ流の意味で, 通常の１台のコンピュータを指す

SIMD (Single Instruction Multiple Data Stream)
単一命令複数データ流の意味で、同一の命令が, 複数のデータの処理を行う.

MISD (Multiple Instruction Single Data Stream) 複数命令単一データ流の意味で, 複数の命令が, 同じデータの処理を行う.

MIMD (Multiple Instruction Mutiple Data Stream) 複数命令複数データ流の意味で、複数の命令が, 複数のデータに対する処理を行う. つまり, すべてのプロセッサユニットが独立な処理が可能である.

これから, さまざまな並列計算機について述べる.

4.1　命令レベル並列コンピュータ

現在次のような方式が考えられている.

スーパスカラ パイプラインの幅を拡げて,1クロックサイクルで複数の命令を同時に実行する方式である.

VLIW スーパスカラ同様, 複数の命令を同時に実行する方式である異なる点は, スーパスカラでは, 並列実行可能な命令群の発行をハードウェアでするのに対して, VLIW では, コンパイラにまかせる.

スーパパイプライン パイプラインの段数を増やすことによって, クロックサイクルを短くするほうほうである.

マルチスレッド アーキテクチャ 全く独立な複数個のプログラムストリームを用意して, これらを並列実行させる方式である.

4.2　SIMD 方式

SIMD(Single Instruciton Multiple Date Stream) 方式では、演算装置は相互結合網で結合され、１つの制御装置で制御される。制御装置から機械命令が放送され、個々の演算装置のデータを使って, 並列に同一演算を実行する。すべての演算装置での処理が終了して初めて、次の命令が可能となる。すなわち命令ごとに同期が取られて実行される。また、演算装置の中に抑止フラグがあり、これが１となっている演算器のみが演算を実行できる構造となっている。

4.2.1　SIMD 方式の特徴

定型処理の高速処理 処理構造が行列などの定型的な処理に対して、高速実行が可能である。処理が非定型的で、制御に分岐が多くなると、処理が遅くなる。例えば、ある１つの演算器に特殊な演算が必要な場合においては、他のすべての演算器を抑止して演算を進める必要が生じ、効率が低下する。

命令レベルでの同期処理 命令実行は１つの命令の完了を待って次命令の発行がなされる。したがって、１つでも遅い命令完了があると、その遅い命令完了を待たなければならない。なるべく同一時間で処理が完了するように、ハードウェアやコンパイラで工夫がなされる。

プログラミングの困難さ 演算装置は相互結合網で結合されているため、結合網が複雑になると、データの各演算器への配置を最適化する必要があり、プログラミングやコンパイル作成は非常に困難になる。

4.2.2　SIMD 方式の現状

SIMD 方式は, マルチプロセッサに同一のプログラムを実行させると SIMD 方式が模擬でき, これを SPMD 方式という. しかし、ニューロコンピュータなど専用目的などには十分利用でき、専用のニューロチップも開発されている。また最近, 画像処理や通信処理用のメディアプロセッサにも, 応用されている.

4.3　演算パイプライン方式

演算装置やメモリアクセス機構を多数のステージに分けて高速処理する方法である。演算は要素演算 E1,E2, ⋯ EL を逐次実行してなされる。これらの要素演算の実行時間が同一でありこれを a としよう。１つのデータに注目してみるとそれが演算器に投入されて処理終了するまでに、aL だけの時間がかかる。しかし、多数のデータに対して要素演算を同時に実行できるのでデータ数がきわめて多い場合には非常に有効である。つまり、a 時間ごとに１つのデータの処理がなされることになる。現在のスーパコンピュータの基本方式として採用されている。

4.3.1　演算パイプライン方式の特徴

高速性 プログラムをうまく構成すれば、通常のコンピュータより１桁以上高速である。

連続性 過去に作成された Fortran プログラムがそのまま利用でき、しかも高速に達成できる。

単純性 ベクトル演算で問題が定式化できれば、最大性能を引き出せる。

汎用性 多様な数値計算分野に適応できる。

単一性 各社で製品化されているハードウェアシステムの構造が同一であり、新しいプログラムを開発するときにユーザ負担が軽い。

高速化困難 パイプラインのマシンサイクルを上げることが困難になってきている。現在の最高速システムでマシンサイクルは2ns となっている。

4.3.2 演算パイプライン方式の現状

パイプライン方式のスーパコンピュータは現在第4世代に到達している。

パイプライン方式のスーパコンピュータは第2世代のCRAY-1でそのアーキテクチャが確定し, 他のアーキテクチャは商業的に不成功となった。そのため、現在のスーパコンピュータはCRAY-1のアーキテクチャに基づいている。

4.4 メモリ共有型マルチプロセッサ方式

超並列計算機JUMP-1がこれにあたる。マイクロプロセッサを相互結合網で結合し、協調させながら動作させ、１つのジョッブの高速実行をはかる。各プロセッサや相互結合網は非同期で動作できるため, プロセッサ間でのデータ授受, 全プロセッサの処理終了通知などのための同期操作が必要となる。

4.4.1 マルチプロセッサ方式の特徴

低価格化 現在,1プロセッサあたりの価格が非常に安くなっているので, 大量のプロセッサをつないでも、それほどコストはかからない。

低電力 マイクロプロセッサはCMOS素子を利用しており, 消費電力は非常に少ない.

並列実行の複雑性 プロセッサは相互結合網でつながっているため、データの分解が不適当であると、通信ばかり多くて性能がほとんどでないことがある。

同期操作 各プロセッサが暴走しないように協調動作のための同期をとらなければならない.

多様性 各メーカーから多様なシステムが開発されているため, あるメーカーのシステムで効率の良いプログラムがあったとしても他のメーカーのシステムではいい性能が発揮できるとは限らない。

4.4.2 共有メモリの構成方式

共有メモリの構成方式には, 以下の方法がある.

集中共有メモリ すべての命令やデータは共有メモリに格納される. この方式の利点は, 各プロセッサからすべてのアドレスが一様にアクセスできる. しかし, その分アクセスに時間がかかり、結合網において競合が起きる可能性が高くなる.

集中共有メモリ (ローカルメモリ付き) 各プロセッサが私的に使用する命令やデータはローカルメモリに格納される. その分, 相互結合網のトラフィックは減少する.

集中共有メモリ (キャッシュ付き) 共有メモリ内の命令やデータが必要に応じて各プロセッサのキャッシュに動的に転送され, 利用される.

分散共有メモリ 各プロセッサにローカルメモリを分散配置したものである。プロセッサに付属するローカルメモリは高速にアクセスできるが、結合網を通して他のプロセッサのローカルメモリにアクセスする場合は, 時間がかかる.

分散共有メモリ (キャッシュ付き) 他のプロセッサに配置された共有メモリの一部を自プロセッサのキャッシュにも動的に取り込み利用する方式である.

また、共有メモリは以下のように分類されることがある。

UMAモデル UMA(Uniform Memory Access) モデルにおいては、共有メモリがすべてのプロセッサから、等距離にあり、アクセス時間が同じである。

NUMAモデル NUMA(Non-Uniform Memory Access) モデルにおいては、アクセス時間が、アドレスによって異なる.

4.4.3 マルチプロセッサの現状

マルチプロセッサは次世代のスーパコンピュータとして注目されている。しかし、, まだ問題が山積みになっている。

4.5 マルチコンピュータ

マルチコンピュータとはコンピュータを LAN(Local Area Network) でつなごうというものである。この方式は非常に安くつき、高速通信なども実現可能になったので, 最近, 非常に注目されている. しかし、現状では無駄があり, 数台のコンピュータを使用しても１台のコンピュータより遅いということも十分有り得る。

4.5.1 マルチコンピュータの特徴

通信速度の遅さ 通信線の速度が非常に遅いので, 通信時間による無駄をできるだけ少なくするために, 大きな単位でプログラムを分割する必要がある. つまり、あまりにも細かい単位で分割すると, 通信に時間をとられ、１台で処理した方が早くなる。

専用の OS の開発 現在の OS はもともとネットワークを意識されて作られた OS ではなく、単独に１台でコンピュータを使用することを目的として作られた OS である. だから、あるユーザプログラムから他のユーザプログラムに１つのメッセージを転送するのに非常に時間がかかる.

信頼性や安全性の配慮 不特定多数のユーザーが使用するため, 他人によるデータ改ざんや停電などによる対策を練っておく必要がある.

4.5.2 マルチコンピュータの現状

マルチコンピュータは非常に安価に構成でき, ATM などの高速通信も可能となったので, 近年非常に注目されている.

5 相互結合網

並列コンピュータの重要な構成要素である相互結合網 (Interconnection Network) について、これからの述べる。

5.1 相互結合網とは

相互結合網とはアレイプロセッサ、パイプライン方式、マルチプロセッサの多数の演算器、プロセッサあるいはメモリを高速に相互結合する網であり、並列コンピュータの性能に重大な影響を及ぼす。演算器、プロセッサなどの相互結合の対象となるものをノード、データ転送路をリンクと呼ぶ。

5.1.1 動作モード

動作モードとしては, 同期モードと非同期モードがある。

5.1.2 制御方式

網内のスイッチ設定などデータ転送に必要な回路を制御する方式として、集中制御方式と分散制御方式がある。

集中制御方式 中央には位置された１つの制御装置によってすべてのスイッチ類の動作が制御される。

分散制御方式 スイッチ自体に制御装置が内蔵されており、各スイッチは自立的に動作する。

5.1.3 交換方式

パケット交換方式 手紙による通信に類似しており、メッセージに宛名をつけて結合網に送り出す。送受信ノード間に物理的なリンクを確立しないで、メッセージを送出できる。１つのパケットの転送が終了すると、使用されたリンクは解放され、他のパケットの利用に使われる。比較的短いメッセージの通信や通信相手が頻繁に替わるような場合に有効である。

回線交換方式 これは電話による通信と類似しており、送受信間ノード間に物理的なリンクを確立するまで時間がかかるが、いったんリンクが確立すると、それが明示的に解放されるまで保持される。したがって、大量のデータの高速転送に適している。また、網の接続形態を長時間変更しないような場合に有効である。

5.1.4 トポロジ

相互結合網の幾何学的形状 (トポロジ) は, 不規則網と規則網があり、規則網には以下の 2 通りに分類できる。

静的網 直接結合網 (direct connection) ともよばれ, あるノードから直接通信できるノードが限定されている.

動的網 間接結合網 (indirect connection) ともよばれ、送受信ノード間にはスイッチ郡がおかれており, 任意のノード間の通信ができる。

参考文献

[1] 舟本氏のプリント

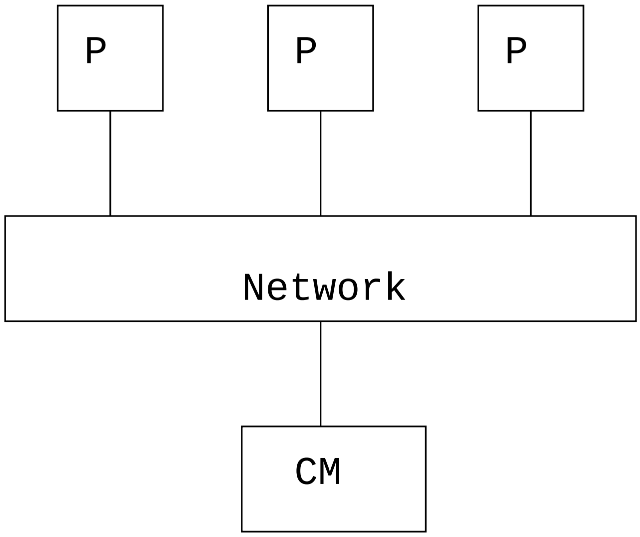

図 1: 集中共有メモリ

[2] 富田 眞治. 並列コンピュータ工学. 昭晃堂.

[3] 富田 眞治. コンピュータアーキテクチャ1. 丸善,1994.

[4] 富田 眞治. 並列計算機構成論. 昭晃堂.1986.

[5] 福永邦雄. 萩原昭夫. コンピュータ通信とネットワーク.

[6] 奥川峻史. 並列計算機アーキテクチャ. コロナ社.1991.

[7] 山本 孝伸. 超並列計算機JUMP-1 における分散共有メモリ管理手法. The Method of Controlling The Distributed Shared Memory on Massibly Parallel System JUMP-1,1997 年 2 月

6 共有メモリの構成方式

共有メモリの構成方式を右図にしめす. 各記号は以下の通り.

P:プロセッサ
CM:共有メモリ
LM:ローカルメモリ
C:キャッシュメモリ

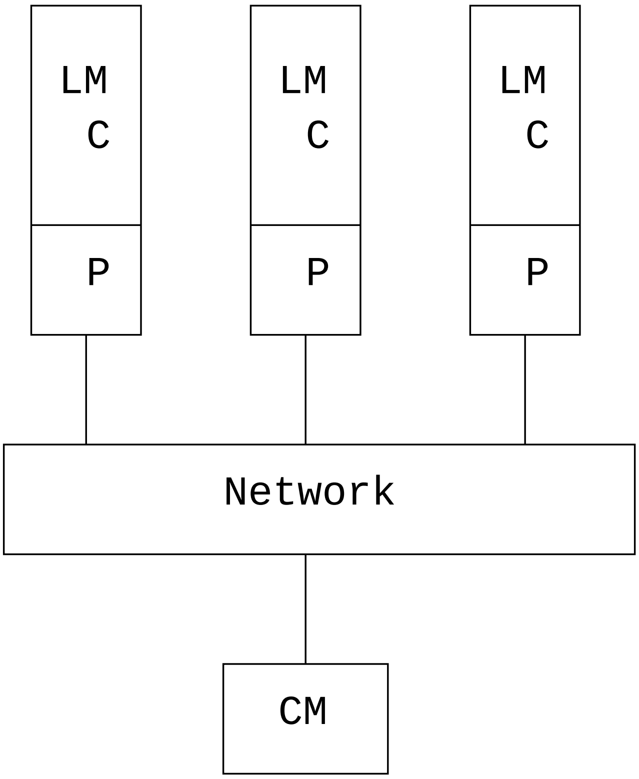

図 2: 集中共有メモリ (ローカルメモリ付き)

7 種々の並列コンピュータ

種々の並列コンピュータを右図に示す。
各記号の意味は以下の通り.

C:制御装置
GM:グローバルメモリ
E:演算装置
P:プロセッサ
LM:ローカルメモリ
S:相互結合網

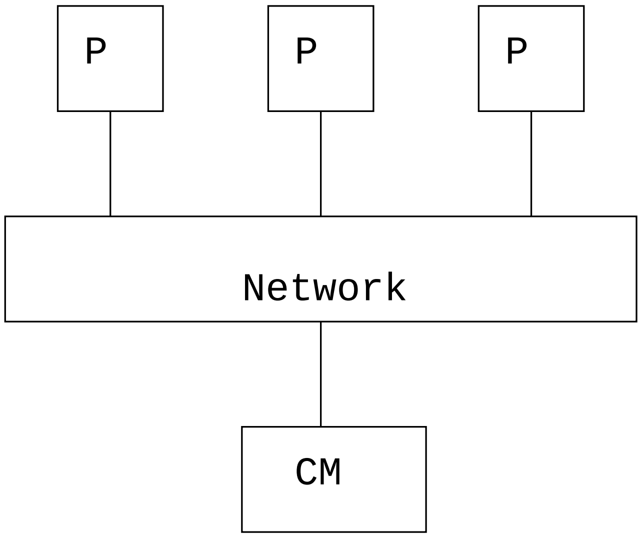

図 3: 集中共有メモリ (キャッシュ付き)

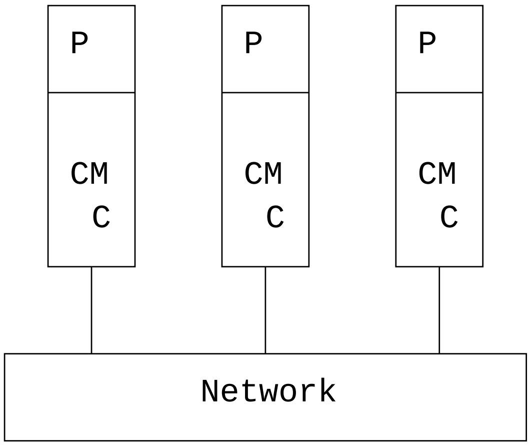

図 5: 集中共有メモリ (キャッシュ付き)

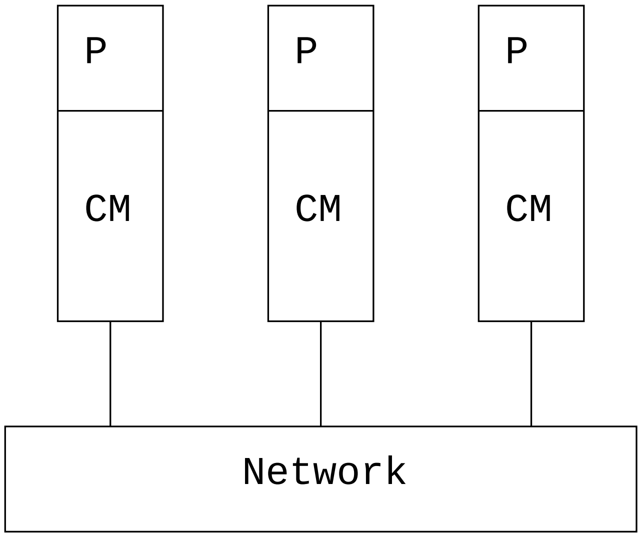

図 4: 分散共有メモリ

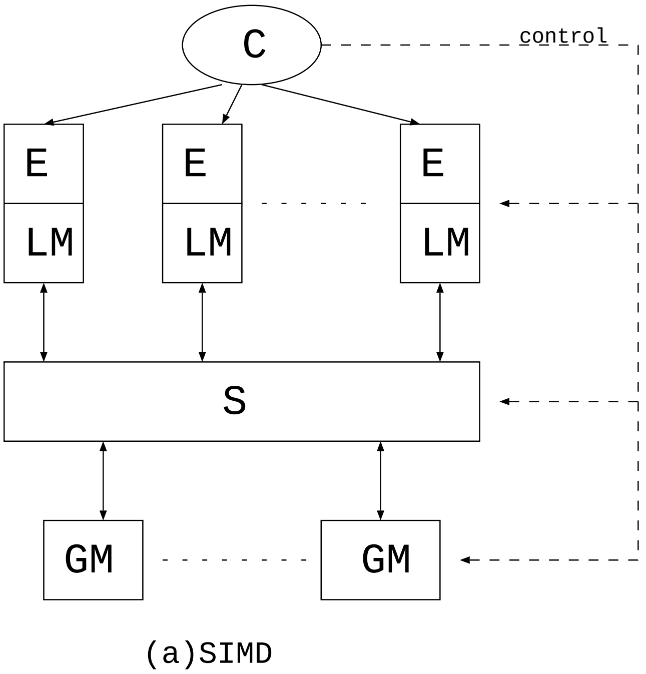

図 6: SIMD 方式

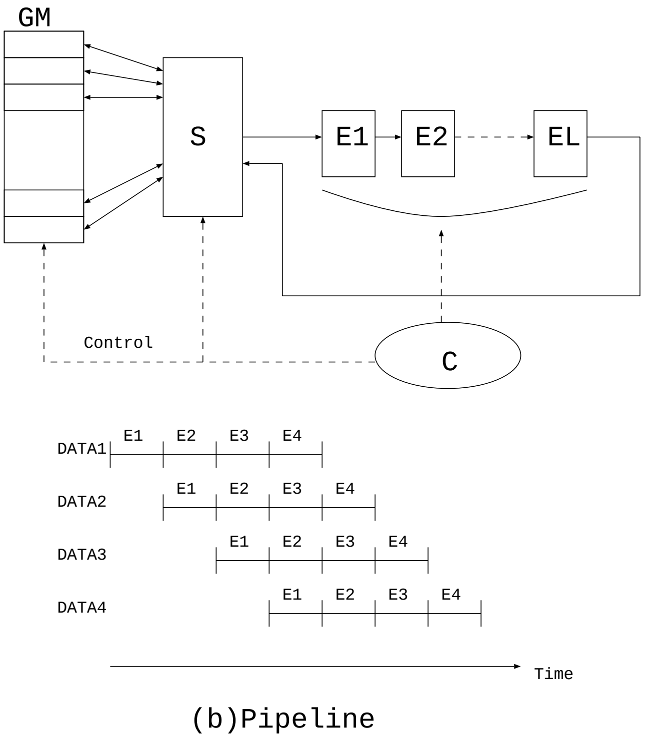

図 7: 演算パイプライン方式

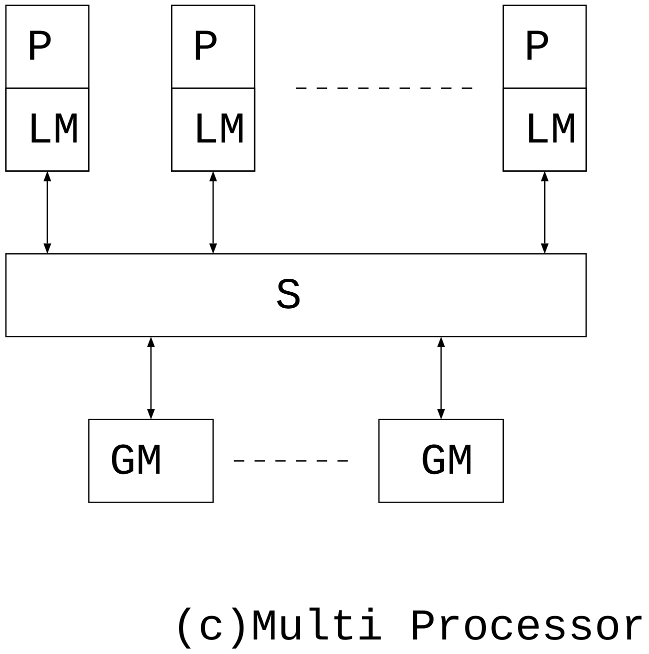

図 8: マルチプロセッサ

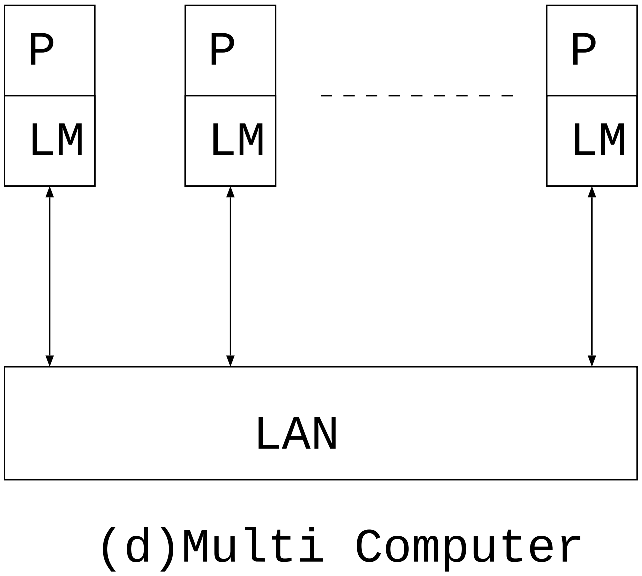

図 9: マルチコンピュータ