

並列処理の歩み

神戸大学計算科学教育センター
小柳義夫

内容

- コンピュータはどうやって 10^{14} も速くなったか？
- アーキテクチャレベルから見た並列処理
 - 並列処理の制御 (Flynn?)
 - データの制御
- 並列処理技術の起源
 - 多くは、1960年代のアメリカ
- プログラマから見た並列処理についての**雑感**

コンピュータの前史

- パスカルの歯車式手回し計算機、加減算
- ライプニッツ: 桁ずらし機構、乗算、除算
「天文学者たちを複雑な計算のために費やす時間を忍耐から救う」
計算は、人間の手を煩わすに値しない労働
- バベジ: 「解析機関」自動式計算機(カードからプログラムを読み取って歯車と「てこ」により計算) — 実現せず。
- Richardsonの夢 (1922)

高性能化技術

コンピュータはどうやって 10^{14} も速くなったか？

- 演算要素の高速化技術
- メモリシステムの高速化技術
- 並列化技術
- 相互接続ネットワークの高速化技術
- ソフトウェア技術

1. 演算要素の高速化技術

(1) 動作周波数の高速化

真空管、トランジスタ、IC、LSI、VLSI

線幅10 nmへ、歩留まり、量子限界

限界：光の波長（すでに遠紫外域）

消費電力

—最近の動向：クロックは上げないほうがよい

未来：新材料（化合物半導体）、超伝導、量子コンピュータ

約 10^7 は周波数の高速化。残りは並列処理。

演算要素の高速化技術

(2) 多数の演算器の搭載と同時実行

演算器は少しの資源で作れる。

問題はどのように制御するか？(→(4))

(3) ベクトル処理および多重ベクトル処理

演算パイプライン: 演算には数クロックかかるが、毎クロックに演算開始

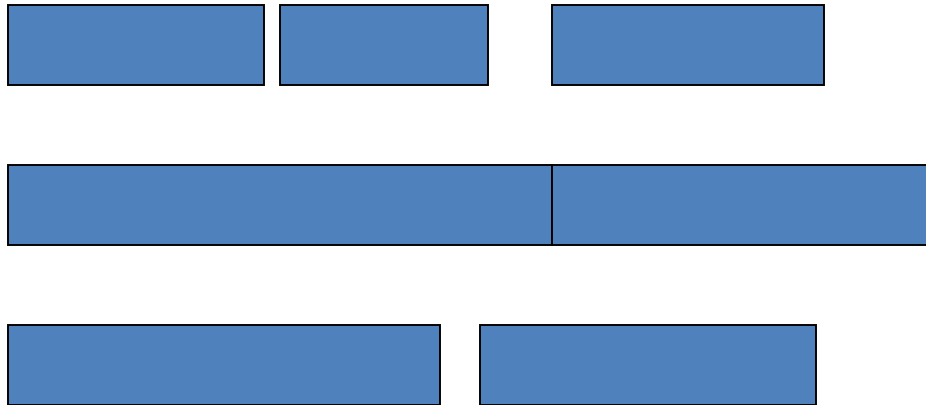
(4) 命令レベル並列化

フォン・ノイマン・アーキテクチャとの整合性を保ちながら、ハードウェアの制御により、複数の命令を並列に発行する。

ベクトル処理も広義には命令レベル並列化。

(4) 命令レベル並列化

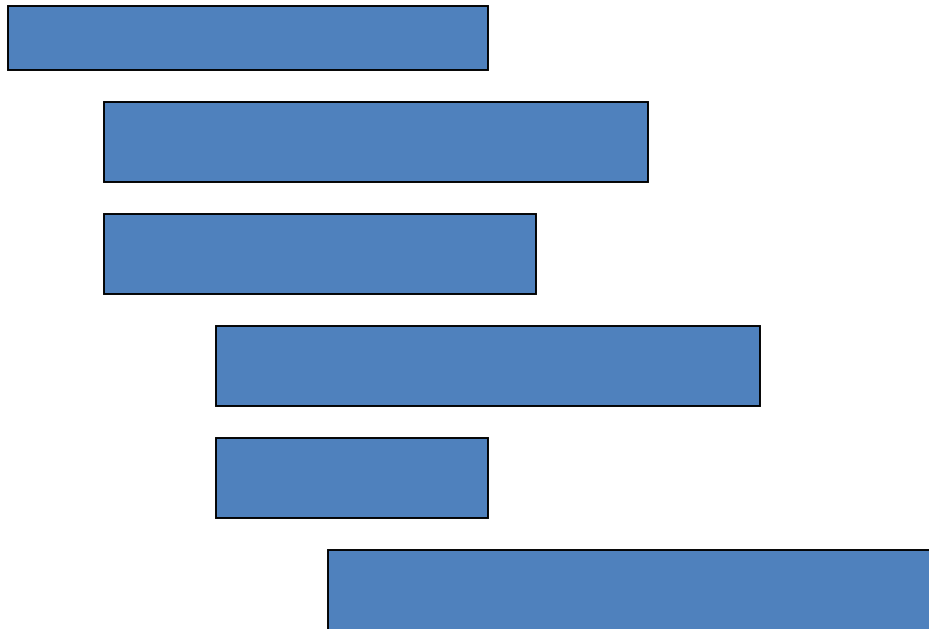
- a) スーパースカラ: 1クロックに複数命令を発行。
CDC6600 (1964) 以来の技術。RISCではi960 (1988)
x86ではPentium (2実行) やP6 (3実行)



実効命令数をどこまで増やせるか。

(4) 命令レベル並列化

b) パイプライン、スーパーパイプライン
スーパースカラと併用。



IBM 7030 (Stretch, 1961)以来。MFでもMicroprocessorでも採用

(4) 命令レベル並列化

c) VLIW (very long instruction word): 1 命令が多くのフィールドから成り、1 命令で複数の処理を起動。

スーパースカラとの違い

コンパイラで並列実行を指定

Josh Fisher (Yale)が提案(1980代)、Multiflow社創立(1984)

富田真治(京大)QA-2 (1983)

Transmeta CrusoeやIntel P6 (Pentium Pro, Pentium II)の内部アーキ
Itaniumアーキテクチャ

(4) 命令レベル並列化

- d) マルチスレッディング、ハイパースレッディング: 1個のプロセッサ上で、同時に複数のスレッドを処理する。あたかも複数チップのように見える。

Denelcore HEP (1982), Tera MTA (1997), MTA-2 (2001)

NetBurst (Pentium 4など)で採用(2000)

- e) SIMD方式: 複数のデータに対して同一の演算命令を並列に実行する。

本来SIMDは、ILLIAC IVのように、単独の並列化技術であったが、マイクロプロセッサでは要素技術。

Pentium IIのMMX (1996)は整数のみ

AMD K6-2の3D Now! (1998)やPentium III のSSE (1999)から

(5) アウトオブオーダー実行機能

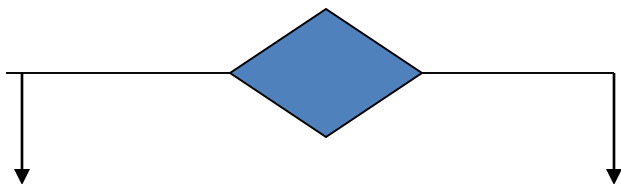
意味的に変わらない範囲で命令の実行順序を(ハードウェアで)変更し、メモリ待ちなどの時間を減らす。

Out-of-order bufferに命令を貯め、依存関係を解析。

- CDC6600 (1964) scoreboarding
- IBM S/360 M91 (1967) register renaming (Tomasuloのアルゴリズム)
- IBM POWER1 (1990)
- Intel P6 (Pentium Pro) (1995)など

(6)分岐予測

条件分岐の行き先を過去の履歴から予測し、
投機的に実行する。



$$A=B-3*C$$

$$A=B+C/D$$

予想した分岐先の演算の、代入の直前まで実行。外れたらやり直す。

IBM 7030 (Stretch, 1961), その後3090 (1985)までなし。

Microprocessorでは多く使われている。

並列処理とは

- 同時または時間的に重複して、複数の「処理」を行う。時分割の並行処理とは違う。
 - ゲートレベル、ビットレベル: 当然
 - 演算器: 並列計算ともいう。
 - BabbageのAnalytical Engineには複数の演算器を計画
 - ENIACには加減算器と乗算器が設置され、並列に動けた
 - プログラム内蔵式に改造したとき、1台しか動かなくなった。
 - Von Neumannレポート: 演算器は高価なので並列処理など非現実的
 - **ノードレベル**: 初期(multics)には耐故障性を重視

並列処理の制御

- SIMD: 全体でProgram Counter 1, Decoder 1
 - 命令のオペランドも同一(例: APE) 当然DM
 - 通常は、オペランドは可変(registerなどで)
- MIMD: ノード毎にProgram Counter 各1
 - 共有メモリ: 排他的制御、キャッシュのコヒーレンシ、同期
 - **なぜThread programmingになったか?** いつから?
 - 元々はprocessだったのか?
 - 分散メモリ: 通信、同期
 - 同期は通信の一種

並列処理とメモリ

- データ (Flynnの分類で欠けている概念)
 - 存在形態 (集中か分散か)
 - アクセス (共有か私有か): 私有なら通信が必要
- 代表的メモリモデル
 - 集中・共有メモリ: いわゆるSymmetric MPなど
 - 分散・私有メモリ: いわゆるmulticomputerなど
 - 分散・共有メモリ: ccNumaなど。SIMDに可能?
 - 集中・私有メモリ: そんなのあり? Master-worker?

共有メモリと制御

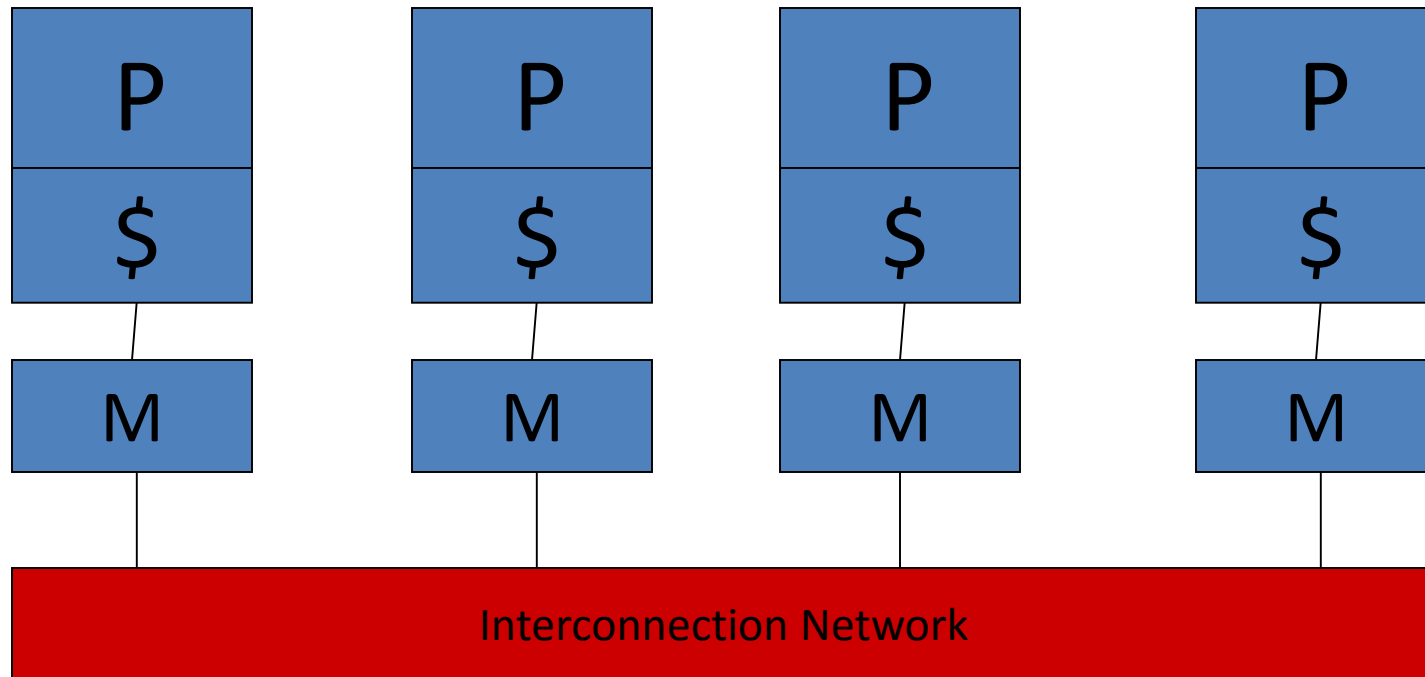
- PRAMは理論モデル、実現困難
- 同期、排他制御
 - SIMDでは心配なし
 - Atomic instructions (compare and swap, fetch and add)
- キャッシュコヒーレンシ:
 - 1983 Goodman : Bus snooping cacheの論文
 - Xbar and directory-base
 - コヒーレンシ無し : C.mmp, T3D/T3E, ...
 - ccNUMA: DASH, KSR, Exemplar, Origin,

分散メモリと通信

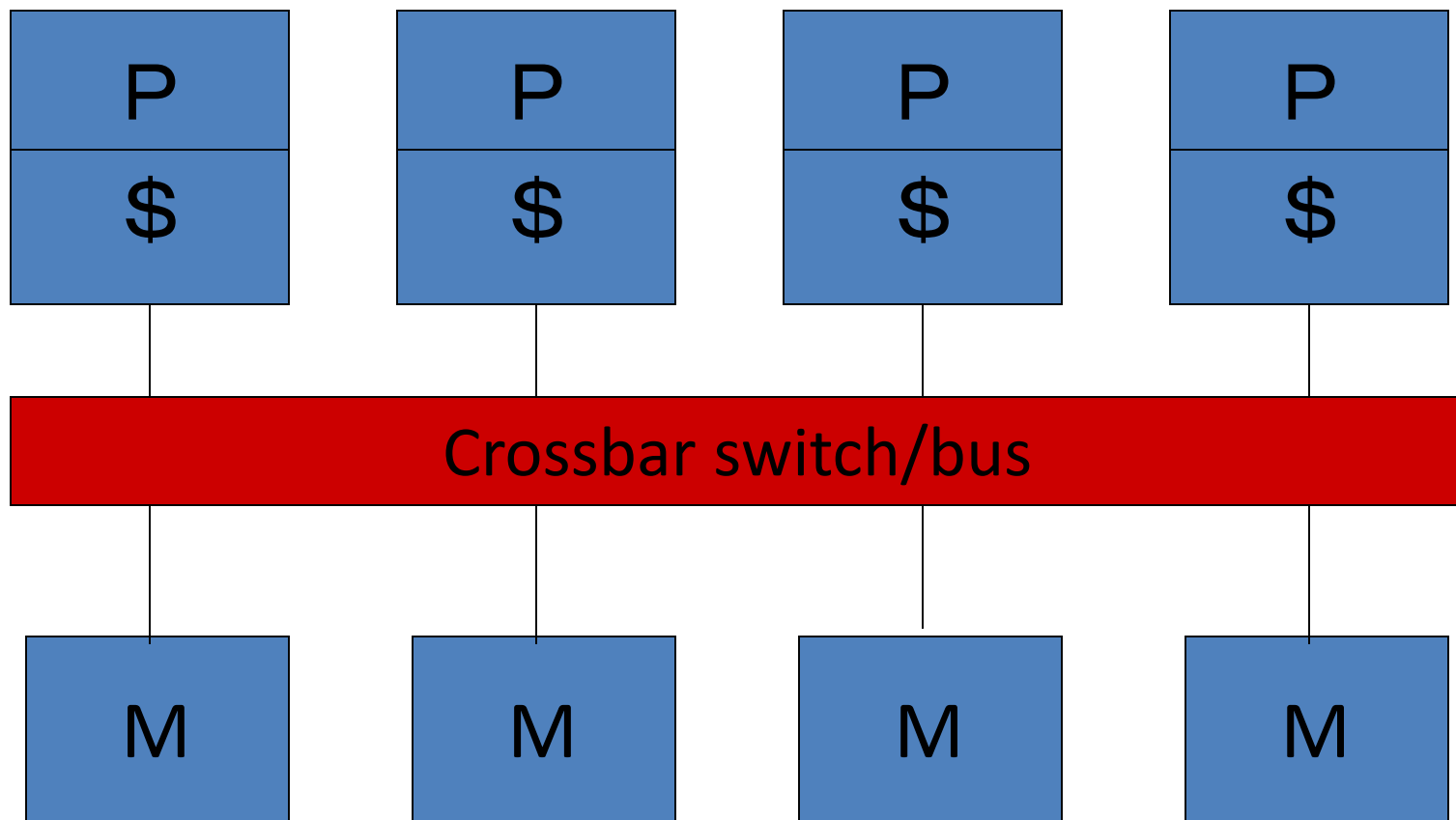
- 部分的共有メモリ
 - 実アドレス空間の一部に共有メモリを割り付ける
 - PAX-32, 128, 32J: 隣接共有メモリ(ステンシル計算用)
 - APE: 隣接ノードのメモリに直接書き込める(SIMD!!)
 - 通信用の小さな共有メモリ(LindaやVPP/Fのモデル)
- OS経由
 - マシン固有: NX/2, NX (Intel), Vertex (nCUBE), CMMD (CM-5), EUI (SP)
 - 共通: Express (Parasoft), p4 (ANL), PARMACS (ANL), Zipcode (LLNL), Chimp (Edinburgh), Chameleon (ANL), PICL(ORNL), Linda (Yale), PVM, MPI
- ユーザレベル通信(put/getなど)

分散メモリ型並列計算機

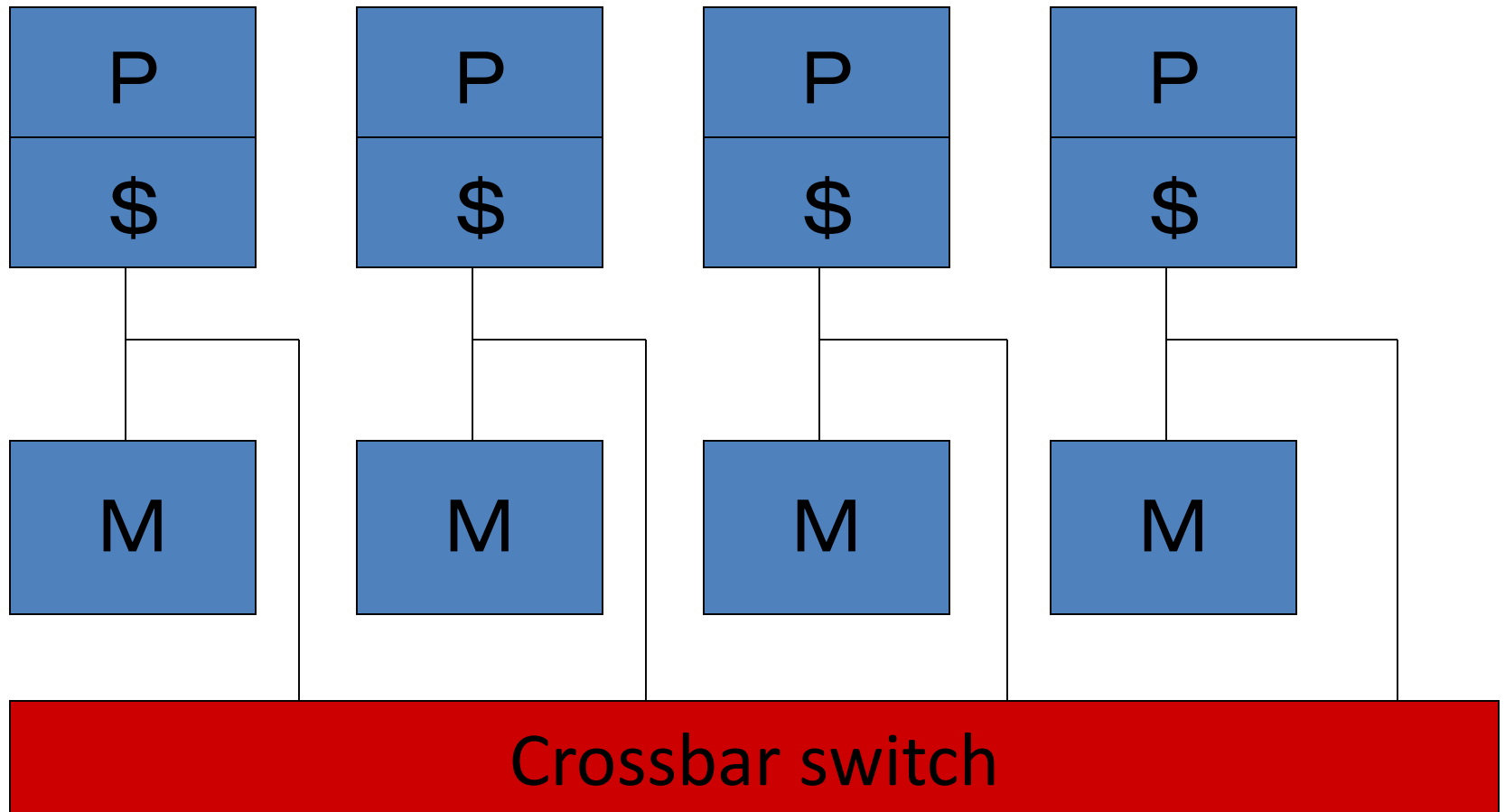
ベクトルの場合は、\$(cache)の代わりにベクトルレジスタ



対称型マルチプロセッサ



分散共有メモリ



並列処理

- 並列処理は自然
 - Babbageのマシンにも複数の演算器
 - Turing machineにはない
 - ENIACには複数の演算器
- なぜ逐次処理
 - Von Neumann レポートでは、演算器は1個
 - 真空管は余りに高速であった(当時)
 - 当時演算器には多くの資源を必要とした

並列化技術

- 並列処理
 - 命令レベル並列化 ((3)(4) で議論)
 - 並列計算機 (SMP, DSM, MPPなど)
- かつては、Flynnの分類などが話題となったが、現在はSIMDなどに名をとどめるのみ。

相互接続ネットワークの高速化

- 昔はトポロジーが議論された: hypercube, torus, mesh, banyan tree,
- 現在、mesh/torus, fat tree, ...
- レイテンシ／バンド幅／距離依存性
(ソフトウェアのオーバーヘッドも馬鹿にならない)
- FIFO性、集合通信、....
- Scalability: one-to-one通信の限界？
- クラスタ: Ether, QsNet, Myrinet, Infiniband,

1960年代のアメリカ

- **科学技術用コンピュータの登場**
 - CDC6600、IBM360/91、ILLIAC IV (契約1964) など
- **基本概念の確立**
 - IBMのMain Frame (1964)
 - Multicsプロジェクト開始(1964, MIT, 完成1969)
 - Mooreの法則(1965)
 - Vector処理の提唱(1965, Senzig and Smith, IBM)
 - Flynnの分類(1966)
 - Amdahlの法則(1967, IBM)
 - Tomasuloのアルゴリズム (1969, IBM)

並列処理の歩み(1960年代)

SIMD: 1958年、UngerがSIMDを提唱

SIMD: 1962年、Solomonプロジェクト中止

?: 1962年、Burroughs社は軍用コンピュータD825を発表。4プロセッサ。
ベースはB5000(1961)

SIMD: 1964年、Illinois大学、ILLIAC IV開発の契約をDARPAと結ぶ

FP: 1964年、CDC6600出荷(10個の並列動作する機能ユニット)

1966年、M.J.Flynn教授(Stanford大学)並列コンピュータを分類

1967年、Gene M. Amdahl (IBM)はAmdahlの法則を発表

SM: 1968年、富士通FACOM 230-60(マルチプロセッサ)

SM: 1969年、Honeywell社はMulticsシステム発表、8プロセッサ

V: 1969年、IBM社はIBM2938 アレイプロセッサ(ベクトル演算器)発売

FP: 1969年、CDC7600出荷

並列処理の歩み(1970年代前半)

scalar: 1970年、日立HITAC 8700発表。

SM: 1971年、Carnegie Mellon大学でC.mmp開発(16台のPDP-11を連結)。共有メモリだがキャッシュ・コヒーレンシなし。10並列を越える初のマシン

V: 1971年、CDC社、STAR-100を発表

SIMD: 1972年、Goodyear Aerospace社、STARAN稼動(1ビットプロセッサ, 4x256)。衛星画像解析

V: 1972年、TI社、ベクトルコンピュータASC完成。30 MF。11月1号機出荷。計6機製造

SM: 1972年、日立HITAC 8700/8800完成(東大では4並列)。

SIMD: 1972年、ILLIAC IVがどうにか稼動(本格稼働は1975)

SM: 1973年、IBM社S/370の上位にマルチプロセッサ方式付加

SIMD: 1973年頃、Burroughs社商用並列コンピュータBSPの開発開始。

V: 1974年、CDC社STAR-100を納入

並列処理の歩み(1970年代後半)

V: 1976年Cray-1をLANLに納入

VLIW: 1976年、FPS社AP-120Bを出荷

SM: 1977年春、Carnegie Mellon大学Cm*(LSI-11が14台)

SIMD + scalar: 1977年、Burroughs社はBSP発表(16 AU, 17 memory banks)。50 MF。1980年中止

DM: 1977年、Siemens社(独)はSMS-201を発表128個の8080をバス結合

SM/DM?: 1977年、ソ連はElbrus 2を製造(10プロセッサ)。初めてのスパコン

V: 1977年、富士通FACOM 230/75 APU

SM: 1978年、富士通FACOM M-200発表(4台のマルチプロセッサ)

V: 1978年、日立、M-180用のIAPを完成(M-200用は1979年)

言語: 1978年、A.HoareがCSPを提案。

DM: 1979年、星野・川合、PACS-9を製作

SIMD: 1979年、ICL社ICL DAPを納入(64x64)

並列処理の歩み(1980年代前半)

DM: 1980年、星野らPAX-32を製作

V: 1980年、CDC社Cyber 205発表

非対称SM: 1981年、BBN Butterfly出荷

全部: 1981年、通産省「スーパーコン大プロ」開始。1989年まで

非対称SM: 1981年、Pyamid社設立

DM: 1981年、G.C. FoxはCosmic Cubeプロジェクト開始。64x8086/87

?: 1982年、ICOT設立。並列推論マシン。1992年まで

PV: 1982年、Cray X-MP/2発表。

V: 1982年、VP-100/200およびS-810発表

DM: 1982年、大村皓一LINKS-1(256台のmaster-worker)

SIMD: 1983年、Goodyear Aerospace社、MPP製造。128x128

SM: 1983年、Elxsi 6400発売

V: 1983年、SX-1/2発表

DM: 1983年、星野らPAC-128を製作

SM: 1983年、Illinois大学でCedarプロジェクト開始(自動並列化コンパイラ)。この頃NY大学ではUltracomputerプロジェクト始まる。

V: 1983年、中国で銀河1号完成

SM: 1984年、Sequent社Balance 8000/21000を発表。最大20プロセッサ

DM: 1984年、星野ら、PAX-32Jを開発

並列処理の歩み(1980年代後半)

DM: 1985年、Transputer発売

PV: 1985年、Cray-2発売

DM: 1985年、nCUBE 10をリリース

言語: 1985年、D. GelernterがLinda言語のアイデアを発表

DM?: 1985年、ドイツでSUPRENUMプロジェクト開始

DM: 1985年、Intel社、iPSC/1を発表

SIMD: 1986年、TM社、CM-1出荷

DM: 1986年、FPS社、T-seriesを発表

DM: 1986年、Meiko社、CS-1を発表

V: 1987年、ETA社ETA-10出荷

DM: 1987年、TM社CM-2を出荷

SIMD: 1990年、MasPar社がMP-1出荷

プログラマから見た並列処理

- 並列ビュー
 - 全システムの処理を記述(global view)
 - 1個のノードの処理を記述(local view)
- Global viewの問題点
 - Von Neumannモデルで並列性記述・抽出
 - データ並列、タスク並列、.....
 - 処理分散(owner-computes rule?)
 - データ分散とそのハンドリング
- Local viewの問題点(プログラマの自己責任)
 - Stencil計算なら簡単(PaxやAPEの原点)

ソフトウェア技術

- 既存の逐次プログラムから指示文の追加で並列化
 - HPF (High Performance Fortran)
 - OpenMP (共有メモリ上)
 - XcalableMP
- 明示的並列化
 - MPI (Message Passing Interface)
 - Thread 並列化 (共有メモリ上)
 - Coarray Fortran (1997, 初めはF--と呼ばれた)

ソフトウェア技術

- 並列化ツール
 - プロファイラ
 - コンパイラ(プリプロセッサ)
- アルゴリズムの並列化
大規模連立方程式の解法など
- モデル化
並列処理向きのモデル化の可能性

おわりに

- 並列アーキテクチャと並列モデル
 - アーキテクチャ: 機能レベルから見たハード
 - 並列プログラミングモデル: ユーザ
- ポスト「京」への問題点
 - Consistency (Lamport), release consistency
 - Reproducibility
 - Fault tolerance
- 多くの課題が残っている