

コンピュータシステムII(1)

情報基盤センター
天野 浩文

1

この科目で学ぶこと

- 単一の計算機ではなく、ネットワーク上に接続された複数の計算機を利用して計算(処理)を行う方式について学ぶ。
 - 概念(並列処理と分散処理)
 - それらを実現するためのシステム構成
 - 複数の計算機を制御するために必須の機構
 - OS(オペレーティングシステム)の基本機能
 - プログラム実行制御方式
 - 同期機構
 - プロセス間通信機構
 - 並列処理・分散処理を支援するための付帯的な機構
 - プログラム作成法

2

授業の進め方・成績評価法

- 教科書+スライドのコピー
 - 教科書は
「並列分散処理」(谷口秀夫著, 電子情報通信学会出版)
 - おおむね教科書に沿った順序で授業を進めるが...
 - 担当教員の判断により, 教科書の記述とやや異なる説明をすることもある。
- 試験・成績評価
 - 数回, 演習を行う。
 - 期末試験も行う。
 - 演習と試験の点数の比率は未定。

3

関連のある科目

- この講義の勉強を進めるうちに, ときどき, 以下の科目で勉強した内容が「なるほど, そういことだったのか!」と思えることがあるかもしれない。
 - オペレーティングシステム(2年後期)
 - コンピュータシステムI(2年後期)
 - コンピュータアーキテクチャI(2年前期)
 - コンピュータアーキテクチャII(3年前期)
- ただし, 上記の科目を履修していること/習得していることが必須というわけではない。

4

計算速度の歴史

～なぜ並列処理が必要なのか～

計算機の歴史についてもっと知りたい場合には、たとえば
<http://virtualmuseum.dlib.vt.edu/>
 のようなサイトを訪れてみるとよい。特に、著作権の関係でこの講義資料に収録できなかったような写真資料を多数閲覧することができる。

5

計算機のスピード(1)

- 「人力の時代」
 - パスカルの歯車式計算機
 - フランスの数学者パスカルが発明(1642年)
 - 何らかの自動機構がある計算機で現存するものでは最古
 - 税務官吏だった父親の仕事を助けるためだったとか
 - ある桁の数字が9から0に上がる(下がる)際に、1つ上(下)の桁の歯車が1上がる(下がる)
 - 加減算のみ

6

計算機のスピード(2)

- ライプニッツの歯車式計算機
 - ドイツの数学者ライプニッツが発明(1674年)
 - パスカルの歯車計算機の改良型
 - 加減算だけでなく、乗除算も自動
 - 18世紀には少数ながら商品として販売
 - 同じ原理の手回し式計算機は、電子計算機が普及するまで用いられていた
- 歯車式計算機の意義
 - それ以前は、計算はすべて人間の手と頭で
 - 暗算、筆算、そろばん、計算尺
 - 桁上げだけではあるが、計算作業の一部を人間から機械へ移しかえた

7

計算機のスピード(3)

- 「機械式計算機の時代」
 - イギリスの数学者バベジの解析機関(1834年)
 - 当時の自動織機の仕組みがヒント
 - 穴のあいたカードで計算手順を「プログラム」できるはずだった
 - 動力は蒸気機関(!)
 - 設計性能は、およそ、1秒あたり1回の加減算、1分あたり1回の乗除算
 - 結局、完成しなかった...機械式自動計算機の実用化は1890年ころ
 - バベジのアイデアの意義
 - 歯車に数をセットしたり、途中の結果を紙に書き写したりするのに人手が介在し、時間がかかるし、間違いも起きやすい
 - 入力装置、出力装置、演算装置、記憶装置を備え、計算結果に応じて場合分けを行うような仕組みまで考えていた

8

計算機のスピード(4)

- 「真空管の時代」
 - 世界初の電子式計算機械
 - ・ 以前は、たいていの本に、「ペンシルベニア大学のENIAC(1946年)が最初のコンピュータ」と書かれていたが...
 - ENIACはとても有名で、下に述べるABCは無名だったが...
 - だが、1970年代の特許裁判でENIACの特許が無効に
 - ・ ABC (Atanasoff- Berry Computer) (1942年)
 - アメリカ・アイオワ州立大学のAtanasoff, Berryが開発
 - 真空管を約300本使用
 - 1秒あたり3.75回の演算ができたと言われている

「電子計算機(コンピュータ)」の定義を「プログラム可変内蔵型の電子式計算機械」とするならば、ABCもENIACのいずれも、それを満たしてはいない。しかし、ここで光を当てたいのは「初期の電子式計算機械の計算速度」なので、あえてそれ以上の議論には立ち入らないことにする。

9

計算機のスピード(5)

- 「トランジスタの時代」
 - 1950年代後半から
 - ・ 日本・通産省電気試験所ETL Mark III(1956年)
 - ・ アメリカ・スペリーランド社UNIVAC SSC(1958年)
 - ・ アメリカ・IBM社IBM-7070(1958年)
 - ETL Mark IIIの演算性能
 - ・ 1秒あたり、約1,800回の加減算、約1,300回の乗算

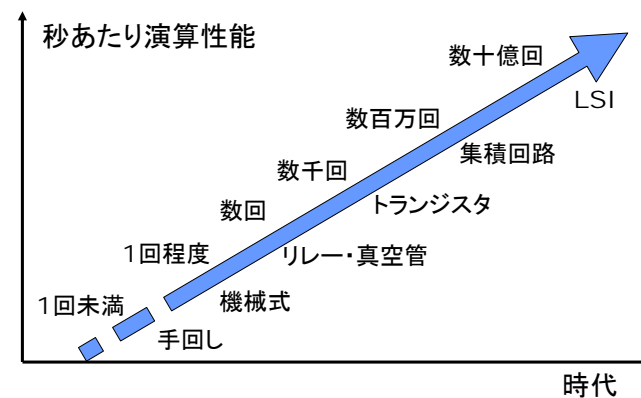
10

計算機のスピード(6)

- 「IC(集積回路)の時代」
 - 多数のトランジスタを1つのチップ上に集積
 - 1960年代から
 - アメリカ・IBM社 System-360シリーズ(1964年～)
 - ・ System-360/195の演算性能(1969年)
 - 1秒あたり約400万回の演算
- 「LSIの時代」
 - 集積度は加速度的に向上
 - 現在のクロック周波数は数GHzのオーダーに
 - ・ 理論的には、1秒あたりクロック周波数×2回(または4回)の演算が可能(1秒あたり数十億回の演算)

11

ここまでのおさらい



12

イギリスの気象学者リチャードソンの夢

- 6時間後の天気を数値計算によって予測しよう！
 - 発案は第一次世界大戦のころ(1922年)
 - 縦横200km・高さ5層に分割した各マス目の気圧変化を計算
 - リチャードソンが1人で1ヶ月かけて手計算でやってみたが、結果は大失敗
 - 当時の気候モデルの欠陥が原因
- 「リチャードソンの夢」
 - 円形劇場に64,000人を配置し、彼の「指揮」に合わせて、隣近所と結果を交換しながら紙と鉛筆で計算すれば、実際の時間の進行と同じくらいのスピードで予測できるはず

13

超高速計算の用途

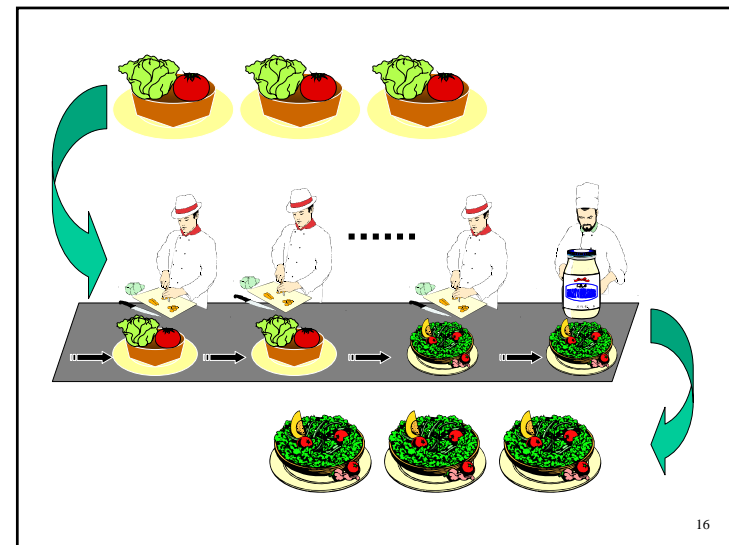
- 最初は軍用(砲弾の飛翔経路の計算や、暗号解読)
- 天気予報(1955年ごろから実用化)
 - 現在では、「数値予報」ぬきに天気予報を語ることはできない。
- 実験・観測が困難(非常に高価)な現象の予測
 - ジェットエンジンのタービン周辺の気流
 - 原子核の構造
 - 自動車の衝突時の変形解析
 - 銀河系どうしの衝突のシミュレーション

14

スーパーコンピュータ

- その時代のコンピュータで、「特に高速のもの」
- 従来は、ベクトル方式(パイプライン方式)が主流
 - 演算装置に、どんどんデータを送り込む
 - 演算を非常に細かな単位に分割
 - 分割された単位を処理する装置を直列に並べて、各処理装置はベルトコンベヤ式(パイプライン式)に処理
 - 演算結果を、演算装置からどんどん吸い出す
- いつの時代でも、最も高価なコンピュータ

15



16

どんどん速くなってくれるはずが...

- ベクトル型のスーパーコンピュータの速度向上はもはや限界に近づきつつある。
 - 「ベルト」の長さ(作業員の数)はあまり増やせない。
 - 演算を非常に細かな単位に分割するには限度がある
 - 「ベルト」の速さももうあまり上げられない。
- 大金を投入して開発しても、もはやほんの少ししか性能が上がらない。
- しかし、同じ性能なら、だんだん安く作れるようになってきた。

17

富士通株式会社の場合

- VP100(1982年)
 - 1秒あたり約2.85億回の演算
- VP2600(1988年)
 - 1秒あたり約50億回の演算
- VPP700(1996年)
 - 1台で1秒あたり約22億回の演算
- VPP5000(1999年)
 - 1台で1秒あたり約96億回の演算

} 1台あたりの性能ではなく、台数で全体の性能をかせぐようになってきた

18

安くなるならもうそれだけでいい？

- 「超高速計算の夢」はどんどんふくれ上がる。
 - もっと大量の仕事を
 - もっと精密な計算を
 - もっと極限的な状況の予測を正確に
- 同じ性能のコンピュータが安くなっていくだけでは、満足してもらえない。
 - 利用者は、他の研究者との研究開発競争に負ける。
 - 計算機メーカーも、ライバル会社に勝てない。

19

他にもっと安いものが大量に...

- マイクロプロセッサ技術の進歩
 - 1つのチップに演算装置をまるまる搭載
 - 集積度・動作速度が急激に向上
 - 大量生産効果により、価格は急降下
- 1つ1つは小さいけれど、それをたくさん集めれば...
- 並列処理とは
 - 複数のコンピュータを同時に用いて、大量の計算を行う
 - 1台あたりの性能は大したことがなくても、全体では相当な性能が出せるはず
- もちろん、世の中はそんなに単純ではない。

20

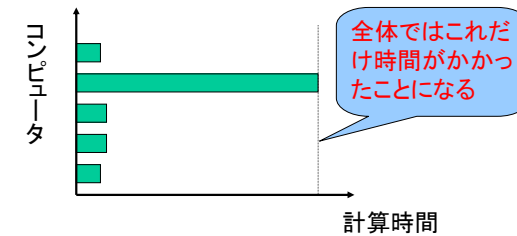
並列処理の問題(1)

- プログラムを書くのがややこしい
 - 並列処理に参加する各コンピュータの間で
 - 互いの進み具合を確認する
 - 他のコンピュータが持っているデータを見せてもらう
- 1台のコンピュータ用に書いたプログラムを自動的に並列化できるとよいのだが...
 - 自動並列化の技術は、まだまだ開発途上

21

並列処理の問題(2)

- 並列処理に参加するコンピュータの負荷が均一でないと、効率が低下する
 - 早く処理の終わったコンピュータが遊んでしまうと、いくら台数が多くても無駄が多い



22

並列処理の問題(3)

- 「船頭多くして舟山に登る」
- 台数が非常に多くなったときに起こりやすい問題
 - 計算している時間よりも、コンピュータどうしが歩調を合わせる(互いの進み具合を確認する)ための処理の時間のほうが長くなる
 - 計算している時間よりも、計算に必要なデータをやりとりする(データを見せ合う)時間のほうが長くなる

23

1. 並列処理と分散処理

注意: 教科書とはやや異なる順序で説明している。

24

並列処理 (parallel processing) とは何か

- 最も単純な定義
 - 複数のプロセッサを同時に用いてひとつの処理を行うこと。
- その「登場」の理由を含めて説明すると...
 - 単一のプロセッサではやりたい仕事をするのに十分な処理性能が得られない場合に、複数のプロセッサを同時に用いて処理を行うこと。
 - 通常は、複数のプロセッサを搭載した単一の計算機(並列計算機)上で実行する。
 - ただし、最近では、そうとばかりも言い切れない(理由は後述)。
 - ただし、複数のプロセッサを用いれば常に高い性能が得られるとは限らない。

25

並列処理が必要になる例

- 非常に巨大な計算をしたい場合
 - 例: 天気予報 (= 巨大な行列計算の集まり)
 - 単一のプロセッサで計算したために明日の天気予報が今日中に出せないようでは役に立たない。
- 非常に多くの場合のすべてについて、ある一定の計算をしたい場合
 - 例: チェスのプログラム
 - こちらの取りうるすべての一手に対して、相手が取りうる対抗策のすべてを洗い出し、その得失を推定して最も良い一手を選ぶ。
 - 「得失」を正確に推定するためには、何十手も先の(厳密には試合が終わるまでの)すべての場合について、得失を計算する必要がある。
 - 単一のプロセッサで計算したために制限時間内に次の一手が計算できないようでは、試合にならない。

26

分散処理 (distributed processing) とは何か

- 最も単純な定義
 - 異なる場所にある複数の計算機を同時に用いてひとつの処理を行うこと。
 - 用語の定義として、「異なる場所にある」という条件は省略すべきでない。
- その「登場」の理由を含めて説明すると...
 - 何かの理由で一カ所にある計算機だけではやりたい仕事が遂行できない場合に、複数の場所に分散した計算機を同時に用いてその処理を遂行すること。

27

分散処理が必要になる例

- ある仕事を完遂するのに必要なデータが、複数の場所に分散している場合
 - 例: 全国各地に倉庫を持つ問屋の物流システム
 - ある大きな注文を処理するのに必要な在庫が社内にあるかどうか確認し、あれば注文を受け付けて、必要なだけの数量を各倉庫から配送させるよう計画を立てる。
 - 倉庫ごとにデータベースが分散していれば、それらにまたがった読み出し処理/書き込み処理が必要になる。
- ある仕事を完遂するのに必要な一連のプログラムが、複数の場所に分散している場合
 - 例: 倉庫は本社に一つだが、営業所・支店・支社がピラミッド状になっている製造販売会社の物流システム
 - 営業所で受けた注文は、支店→支社→本社の順に一連のプログラムを経由して転送され、本社の決裁終了後、逆に営業所まで伝達されなければ、発送の手配が完了しない。
- データとプログラムの両方が分散するようなケースもありうる。

28

クライアント・サーバ・システム

- 分散処理システムの一例
 - クライアントは、自分でできない(やらない)仕事を、サーバに依頼する。
 - プログラム記述の形態としては、いろいろある。
 - データを送信し処理を依頼して結果を受け取る一連の通信制御手順を陽に記述する。
 - データの送受信について、遠隔ファイルやメモリを共有する機構を利用する。
 - サーバ上の手続きを呼び出して結果を取得するための特別な「しかけ」(遠隔手続き呼び出し, remote procedure call, RPC)を用いる。

29

並列処理・分散処理の難しさ

- 並列処理・分散処理の特徴
 - 異なるプロセッサ(計算機)の上で進んでいく複数の処理が、最初から最後まで完全に独立ということはない。
 - ときどき、必要なデータを他のプロセッサ(計算機)から取得する、または、他のプロセッサ(計算機)にデータを送る必要がある。
 - ときどき、他のプロセッサ(計算機)の処理が終わるまで待ち合わせる必要がある。
- 逐次型処理とは異なり...
 - プロセッサ(計算機)間で、データの送受信や処理の依頼のための通信を行う必要がある。
 - プロセッサ(計算機)間で、処理の進み具合を調整する必要がある。

30

並列処理に用いられるハードウェアの例

- 並列処理用に強化された特殊な計算機(詳細は2節を参照)
 - 複数のプロセッサ
 - メモリは個別に持つ場合と全体で共有する場合がある。
 - 相互結合網(interconnection network または interconnect)
 - プロセッサやメモリを接続し相互に通信させるための「通信路」で、さまざまなトポロジ(topology, 位相幾何学的形状)を持つものがある。
 - 非常に短い時間で通信を終えられるように工夫された特殊なハードウェアを備えている(ものが多い)。

31

並列処理に用いられるソフトウェアの例

- OS(オペレーティングシステム, operating system)
 - 並列計算機用のOSが搭載されることが多い。
 - 並列計算機のためのシステム制御, 実行管理などの機能を持つ。
 - 並列計算機を効率的に利用するためのスケジューラも持つ。
- 応用プログラム作成の支援法はいろいろ
 - (自動)並列化コンパイラ
 - プログラムが意識しなくとも、単一プロセッサ用に書かれた逐次型プログラムが自動的に並列化される。
 - 並列処理記述言語(並列プログラミング言語)
 - 並列プログラムであることを意識して、専用の言語で新たに記述。
 - 並列処理用ライブラリ
 - 逐次型プログラム(の複製)が並列プログラムとして動作できるように、特殊な手続き呼び出しを追加して改造する。

32

分散処理に用いられるハードウェアの例

- インターネットで接続された、遠隔地に分散する複数の計算機
 - 特別な呼称はないが、仮に「広域分散処理システム」と呼んでおこう。
 - 計算機の構成(プロセッサ種別, メモリ量など)は異なることが多い。
 - 遠隔地にあるプロセッサを連携させたり制御したりする特別なハードウェアは持たない。
 - 分散処理に使用しないときでも, それぞれは単体の計算機として動作する。
 - それぞれの計算機のメモリは共有されない。
 - 並列計算機のような特別な通信用ハードウェアは持たない。

33

分散処理に用いられるソフトウェアの例

- OS
 - 基本は, それぞれの計算機に搭載されている通常のもの。
 - 他の計算機とのデータ交換や処理依頼のために, 通信機構やファイル機構が強化されていることがある。
 - 分散した計算機間でメモリを共有するためのソフトウェアの機構(分散共有メモリ, distributed shared memory, DSM)を備えていることも。
- 応用プログラム
 - 逐次型プログラム(の集まり)が分散プログラムとして動作できるように, 特殊な手続き呼び出しを追加して改造する。
 - 分散処理であることを意識して記述しなければならない。

34

利用形態から見た並列処理と分散処理の比較

- ジョブ: 利用者が出した処理要求のまとめり
 - 単一のプログラムのことも, 一連のプログラムのこともある。
 - 長いものも, 短いものもある。
 - ジョブを投入したら, 利用者は処理が終わるまで待っている。
- 並列処理の典型的な利用形態 (教科書の表1.2の並列処理の①)
 - 処理の高速化のために必要であれば, 並列計算機全体を単一のジョブに占有させることがある。
- 分散処理の典型的な利用形態 (教科書の表1.2の分散処理の②)
 - 単一のジョブのために各計算機を同時に占有することは難しい。動いていないものもあるかも知れない。
 - 他の計算機での処理の結果が届くまでの時間が少々長くなってもよいようなプログラムの書き方をしたほうがよい。

35

『利用形態』に関する教科書の解説への補足(異論?)

- 表1.2の「並列処理」の②
「ハードウェア情報(プロセッサ数やメモリ量など)を利用したプログラム記述を行う」
 - 確かに, 負荷分散による性能向上につながる可能性はある。
 - しかし, 他の並列計算機で動作しなくなる(あるいは, 性能が出なくなる)原因にもなりやすいので, 一長一短あり。
- 表1.2の「分散処理」の①
「各計算機に関する情報(計算機の場所やプロセッサ種別など)を意識しない形で利用する」
 - 可能であれば, 確かに, 理想的である。
 - しかし, そのためには特別な「しかけ」が別に必要であり, 残念ながら, まだまだ完成された技術とは言い難い。

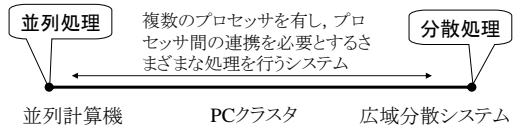
36

並列処理と分散処理は決して「別もの」ではない

- ハードウェアの境目はだんだん曖昧になりつつある
 - 例: PCクラスタ

1Gbit/s
以上で
なくとも
よい。

- 同種のPCを複数台、ネットワークスイッチ(汎用のもので可)で接続
- ハードウェア的には分散処理向けに見えるが、そんなに遠隔地ではない(同一室内, 同一建物内)
- しかし, 並列処理用ライブラリを使うことで, 並列処理にも利用可能



37

まとめ

- 計算速度の歴史
 - 「機械にやらせる計算」の速度の歴史
 - なぜ並列処理が必要なのか
- 並列処理とは何か
- 分散処理とは何か
- 並列処理・分散処理に用いられるハードウェア・ソフトウェア
 - 詳細は次回に

38