

# オペレーティングシステム

第10回(2009.06.18)

分散オペレーティングシステム

# 参考資料

- 分散オペレーティングシステム
  - -- UNIXの次にくるもの --
  - 共立出版株式会社
  - 前川守・所真理雄・清水謙多郎編
  - ISBN4-320-02570-9 (¥6,180)

# ホンダの車は「協力」しあう！？

- インターナビ・フローティングカーシステム
  - 実際に走っている車が情報を提供する。
  - 現在の道路情報を更新できる。
  - ホンダのWEBページから
    - 「あらかじめ設定された道路区間をメンバーの車が走行した際、その所要時間をセンターにアップロードし、」
  - 「アップロード」 ⇒ センター・サービス
    - クライアント・サーバ型？

# 並列処理⇒分散処理

- プロセスの分割
  - 一つのCPUに、複数のコアがある場合
    - Dual CORE
    - CPU内部の配線で、通信
  - 一つの筐体内部に複数のCPUがある場合
    - マルチCPU
    - バスを経由して通信
  - ネットワークを経由したコンピュータ連携
    - ネットワーク経由で通信、分散並列処理
- ⇒ 分散環境にふさわしいOS

# 情報(データ)をどう共有するか？

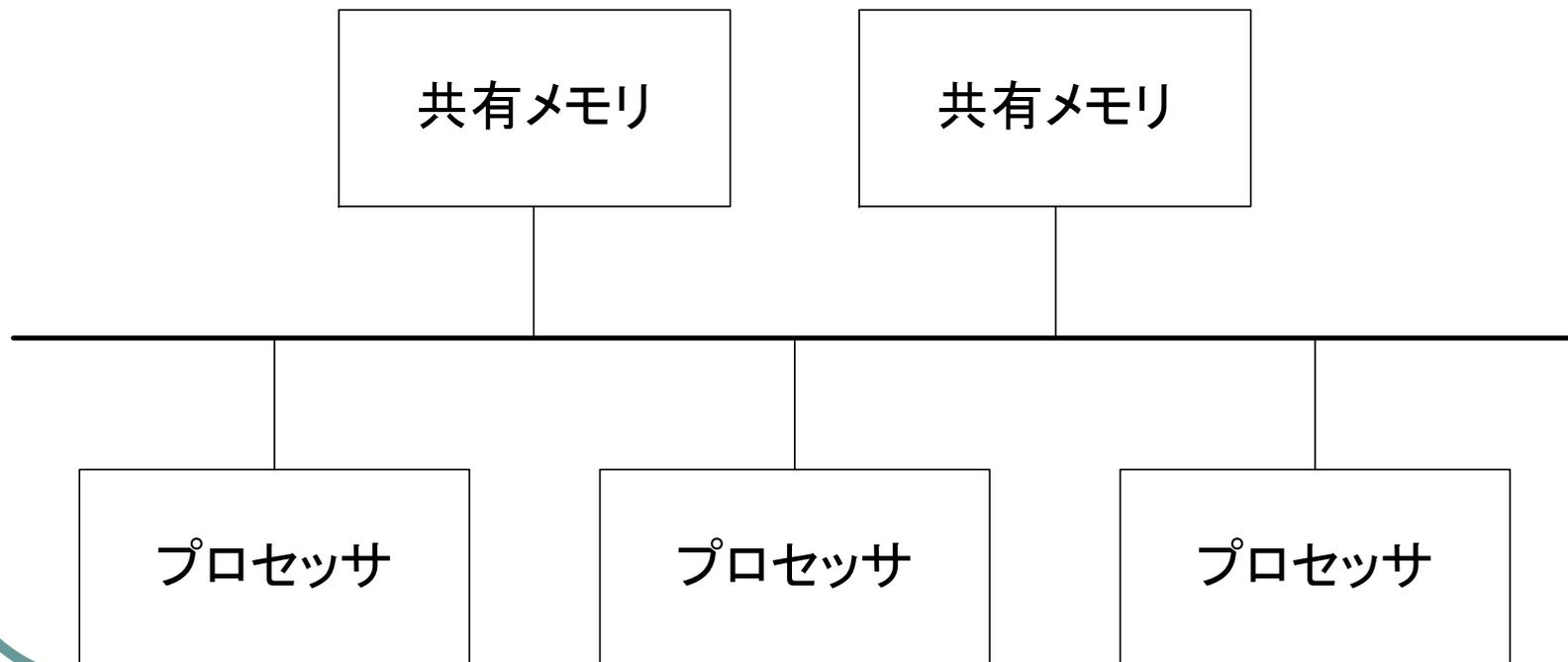
- データへのアクセス方法が「鍵」
  - メモリや、ファイルなどの「情報」の共有方法
    - 「分散」の方法の根幹になる。
- 三つのキーポイント
  - メモリの共有方法
  - ファイルの共有方法
  - 通信の方式

# 分散と結合の程度

- 共有メモリ型マルチプロセッサシステム
  - 密結合型マルチプロセッサシステム
    - 共有メモリへのアクセス
      - 排他制御
      - ロック機構
- 分散メモリ型マルチプロセッサシステム
  - 疎結合型マルチプロセッサシステム

# UMA

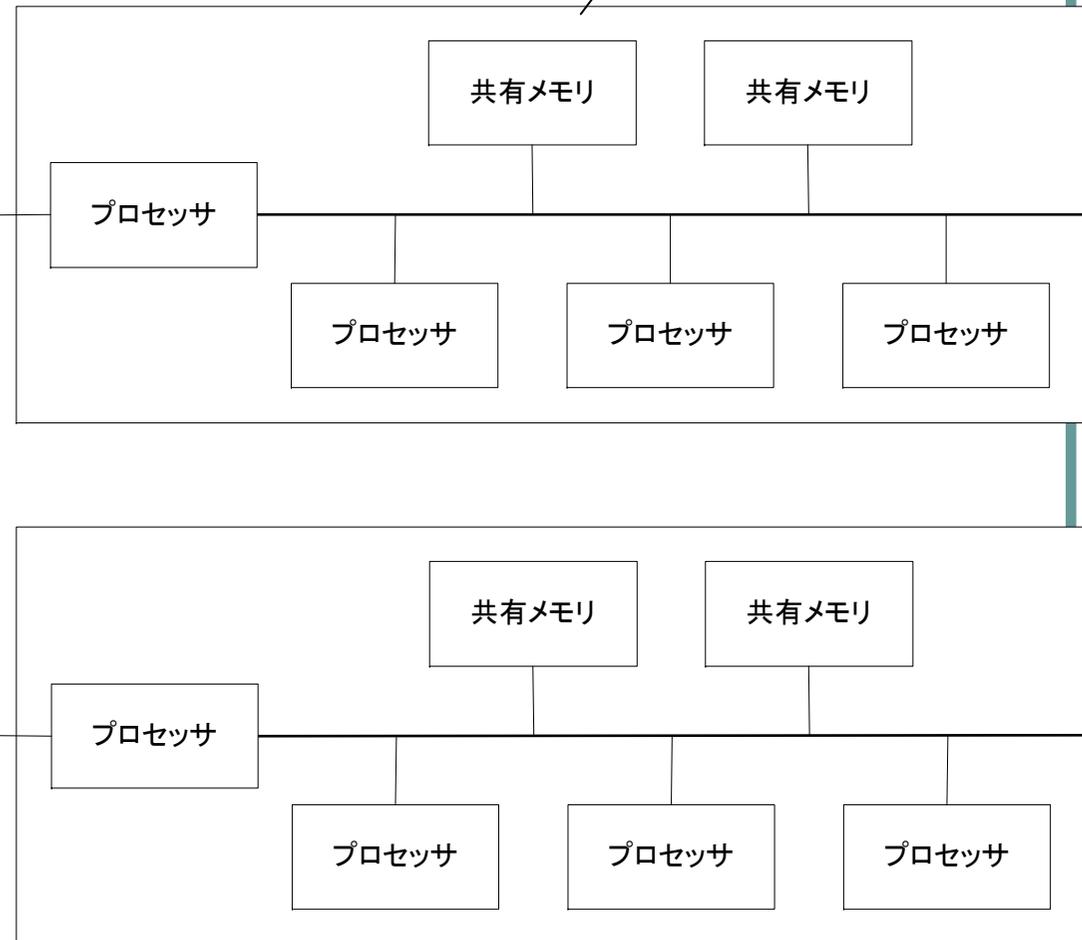
- 共通のシステムバスで結合
- Uniform Memory Access



# NUMA

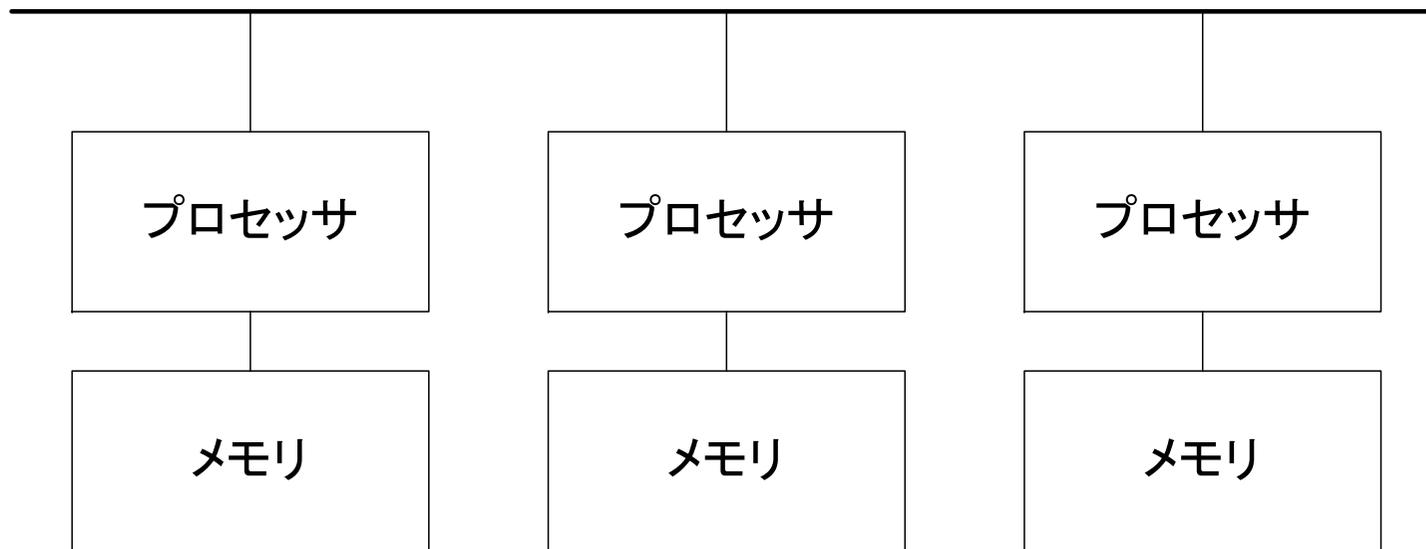
クラスタ

- 共有メモリに対するアクセスが不均一
- 階層メモリを持つものなど
- Non Uniform Memory Access



# NORMA

- No Remote Memory Access
- 疎結合型マルチプロセッサシステム
- 共有メモリを持たない



# プロセス間通信の方式

- 疎結合型のシステム
  - メッセージ伝達 (Message Passing) 以外に、データ共有の手法がない。
- 密結合型のシステム
  - 共有メモリを用いて、連絡できる。
  - 同時に、Message Passingを実施しても良い

# 間接通信

- Interfaceを介して行う
  - Channel
  - Path
  - Mailbox
  - Exchange
  - Link
  - Port
- 様々な名称を持つが、機能は
  - send / receive

# Client – Server型

- クライアント / サーバ型の通信
  - クライアント: 要求する側
  - サーバ: サービスを提供する側
- WEBサービスは、クライアント/サーバ型
  - WEBサーバ
  - アプリケーションサーバ
  - データサーバ

# プログラムをどこに置くか

- 必要なプログラムを、全て予めローカルに持っている
  - ⇒ 記憶域の全体効率は良くない
  - 分散システムというより、「関係動作」しているスタンドアローンのマシン群
- ランタイムに、必要な「プログラム」(手続き; Procedure)を、随時取得できる。
  - 遠隔手続き呼び出し
  - RPC, Remote Procedure Call

# RPCとスタブ

- スタブ(stub)って何？
  - 未完成の部品を使用してプログラム開発する際の、dummyのテストモジュール
- RPCがstubを使う。
  - 実際には遠隔地にあるプロシージャが、ローカルに存在しているかのように見せる役割
    - クライアントスタブ
  - 実際には遠隔地から呼び出された呼び出し側プロシージャが、ローカルに存在しているように見せる役割
    - サーバスタブ

# スループットって何？

- 教科書 P191
- システムの単位時間あたりの処理量
  - (件数/時間)
  - 件数 = ジョブ数 または トランザクション数
- 負荷分散と負荷均衡
  - 負荷分散: Load Sharing
  - 負荷均衡: Load Balancing

# ターンアラウンドタイム

- システムに要求が届いてから、反応が返るまでの時間。
  - → 反応時間
  - Turn Around Time
- リアルタイム処理系では特に要求が厳しい
- 分散システムの個々のTurn Around Timeの数字が出ていないと、全体のシステムの性能が予測できない。

# 分散ファイルシステム

- NFSやRFS

- NFS: Network File System
- RFS: Remote File Sharing
  - LANに接続された、他のPCのファイルを編集できる。
  - ⇒ Windowsで実現済み

- ノードをマウントする

- 使いたいファイルがどこにあるか、意識しなくてすむ。

# CVS – 同期型のファイルサーバとして

- CVS: Concurrent Versions System
  - 平行バージョンシステム
  - 最新バージョンと、現在のバージョンの差分だけをダウンロードできる。
- 複数のメンバーが、大規模プロジェクトを同時進行で実行する際に、全員が、必要最小限の操作で、差分を入手できる。

# ユビキタスな環境

- ユビキタスって何？
  - Ubiquitous : どこにでもある
- 「自分のコンピュータ」という感覚がなくなる。
  - (たぶん)新しいテーマ
    - 自分の携帯電話と、拠点にあるコンピュータの連携
    - 携帯電話で、(自分のサーバの)プロセスを実行して、結果を受け取る。
  - こうした携帯環境を実現する新しいOSのコンセプトは、(たぶん)携帯専用に必要なになる。
    - ⇒ 皆さんの仕事(テーマ)です。

# まとめ

- 分散システムは、CPU(処理本体)が複数存在し、リモート(遠隔)で互いに通信しあうシステムである。
- 分散オペレーティングシステムは、分散システムの性能の向上のために用いられる
- メモリ管理、ファイルシステムのそれぞれが、分散システムとして構築されている。
- プロセス間の通信手法が問題となる。
- 負荷を分散させ、あるいは均衡させて、システムとしてのスループットを向上させる。