分散システムの概要

分散システム

2011年12月6日

建部修見

導入（1）

- コンピュータシステムの進化
  - 1945～1985頃までは，巨大で高価であった
  - ミニコンですら数千万円，単独で利用された
- 80年半ばからの大きな技術革新
  - microprocessorの発達
    - 10億円で毎秒1命令から10万円で毎秒10億命令
    - コストパフォーマンスは10^{13}向上
  - コンピュータネットワークの発達
    - LANにより数百のコンピュータが100Mbps～10Gbpsで接続
    - WANにより地球上の数百万のコンピュータが64Kbps～100Gbpsで接続

導入（2）

- 集中システム（centralized systems）から
  - 単一のコンピュータ
  - 周辺機器
  - 遠隔端末
- 分散システム（distributed systems）へ
  - 高速ネットワークで接続された多数のコンピュータからなるシステム

分散システムの定義

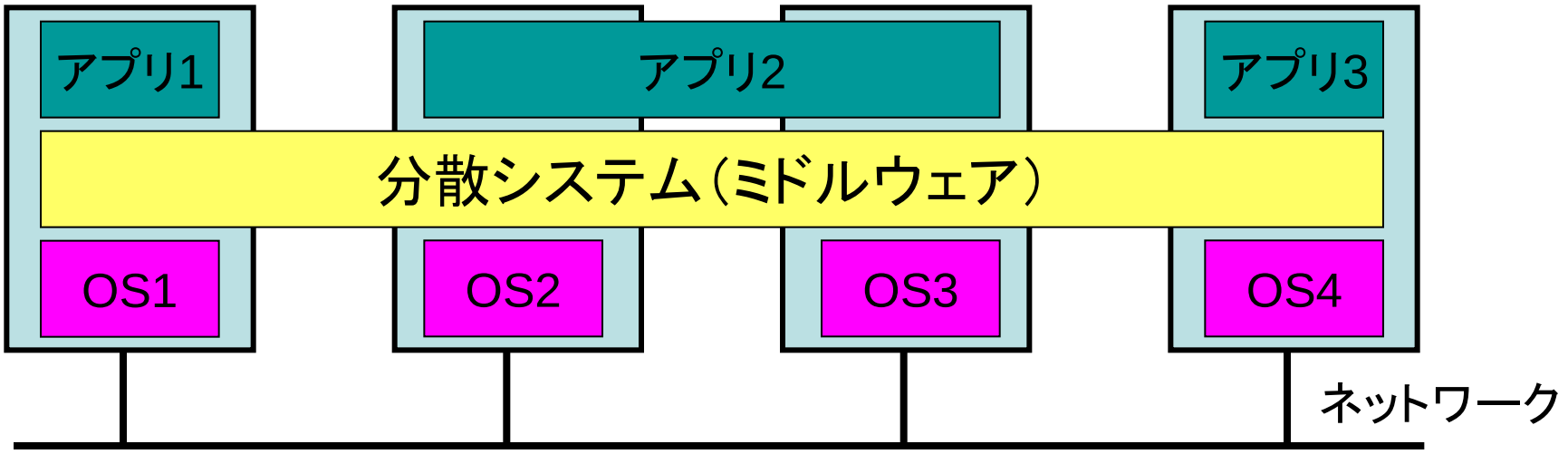
- **単一**の(single)**一貫**した(coherent)システムと**見なせる，独立した**(independent)コンピュータの集合
  - 自立(autonomous)したコンピュータからなる
  - 利用者(プログラム)は単一システムと見なせる
- 独立したコンピュータの**協調**が必要
  - どのように協調するかが分散システム構築のポイント
  - コンピュータの種別(性能，能力)は問わない
    - スパコンからセンサーまで

分散システムの*特徴*

- コンピュータの差，コンピュータ間の通信はユーザから**隠蔽**
- **一貫**した（consistent）**均一**な（uniform）なアクセス
- 原理的には**拡張**，**スケール**が比較的容易
  - 独立したコンピュータだから
  - ただし，実際にどのように構成しているかは隠蔽する必要がある
- **いつでも**利用可能（continuously available）
  - 一部が故障した，修理された，追加されたとしても利用者には気づかせない

分散システムのソフトウェア階層の例

- 計算機の非均質性（heterogeneity）とネットワークに対応するため，アプリケーションとローカルOSの間のミドルウェアとしてしばしば実装される



- 各アプリケーションに同一のインターフェースを提供
- 分散アプリケーションが相互に通信する手段を提供
- ハードウェア，OSの違いを可能な限り隠蔽

4つの目標

- **簡単**に利用可能
- **分散**していることを十分**隠蔽**
- **オープン**
- **スケーラブル**

簡単に利用可能に

- 遠隔の資源を簡単に利用できるように
  - 制御され，効率的に共有
- 資源：プリンタ，コンピュータ，ストレージ，データ，ファイル，Webページ，ネットワーク，…
- 資源共有の理由
  - 経済的なもの
  - それぞれにプリンタを購入し維持するより複数ユーザで共有した方が安い
  - スパコン，高性能ストレージ，イメージセッタなど高価な機器の共有

資源共有の例

- インターネットプロトコルにより共同作業，情報交換が容易に
  - 単純な標準プロトコルにより，ファイル，メール，音楽，ビデオ交換が容易に
  - グループウェアにより地理的に離れたグループでの共同編集，テレコンなどが可能に
  - お店に行くことなしにネット購入も可能に
- セキュリティが問題に
  - 通信の盗聴や介入に対する対策が必要
  - パスワードなど暗号化されずクリアテキストで送信，サーバに格納されることも
  - カード保有者である証拠を見せることなくカード番号で購入する
  - ユーザの好みを知るためなど通信履歴の追跡を許可なく行う
  - スパムメールなど無用の通信からの対策

分散透明性（Distribution Transparency）

- 目標
  - プロセスや資源が複数のコンピュータに分散していることを隠すこと
- 単一のコンピュータのように振る舞う＝**透明**（Transparent）である

透明性の種類 [ISO, 1995]

透明性	意味
アクセス	データ表現の違い，データアクセス方法の違いを隠蔽（エンディアンなど）
位置	資源がどこにあるか隠蔽。ネーミングが鍵（位置が名前に含まれない論理名）
移動	資源が移動したことを隠蔽
再配置	使用中に移動したことを隠蔽
複製	資源の複製があることを隠蔽
並行性	複数ユーザで共有されていることを隠蔽。一貫性を保つ
障害	障害と復帰を隠蔽

透明性の程度

- 完全な透明性の確保はいいことではない！
- 時差，通信遅延を隠すことはできない
  - 光速は越えられない
- 透明性の程度とシステムの性能は**トレードオフ**の関係
  - 最終的に失敗する前に何度もリトライ。そのため反応が遅くなる
  - 大陸をまたぐ複製。更新を伝えるだけで秒オーダ
- 組込やユビキタスシステムでは**分散を見せる**ほうがよい
  - ノートPCでプリントアウトする場合，国が異なる本部の空いているプリンタより近くの忙しいプリンタのほうが良い
- 分散を見せることにより，より効率的な利用も可能
- 設計では完全な透明性は良い目標だがそのコストは恐ろしく大きい。**性能**と**分かり易さ**は重要。

Openness

- オープンな分散システム＝標準規格に従いサービスを提供するシステム
  - 標準通信プロトコル（フォーマット，内容，メッセージの意味）
- 分散システムはインターフェースで定義されることが多い
  - サービスのインターフェース記述にIDL（Interface Definition Language）を利用
  - サービスの名前，引数の型，返値，例外
- オープンシステムの目標
  - **相互運用性**（Interoperability）＝異なる実装でも仕様に従っていればお互いに利用可能
  - **ポータビリティ**（Portability）＝システムA用の実装をシステムBで無修正のまま利用
  - **拡張性**（Extensibility）＝コンポーネント追加，変更が容易

ポリシの分離

- システムの柔軟性のためには，比較的**小さく，簡単に変更可能**な**コンポーネント**で分散システムを構成するのがよい
  - システム内部のインターフェースの定義と相互作用の記述が必要
- 一つの巨大プログラムで実装されるモノリシックなアプローチは，コンポーネントの変更が難しく，クローズなシステムとなりやすい（一般論）
- 分散システムの変更要求は，しばしばコンポーネントのポリシの変更
  - 内容によるWebキャッシュの例：列車の時刻表は残したいが刻々と変わる高速道路の現在の交通状況は残したくない
- ポリシとメカニズムの分離が必要
  - 様々なパラメータ，あるいはポリシの記述を可能に

スケーラビリティ

- 最も重要なデザインゴール
- 三つの次元（Neumann，1994）
  - **サイズ**。利用者や資源を追加可能
  - **距離**。地理的に離れて利用可能
  - **管理**。独立した組織をまたいでも管理が容易
- ただし，上記の一，二にスケーラブルなシステムは，スケールアップすると**性能**に**影響**することがある

スケーラビリティの問題(1)

- サイズに関するスケーラビリティを考えた場合
  - 集中型サービス(多数ユーザに対応できない)
  - 集中型データ(大規模データに対応できない)
  - 集中型アルゴリズム(分散データの収集，配布で破綻)

  は避けなければならない
- 非集中型アルゴリズムは以下の点で集中型と異なる
  - どのノードもシステム**全体の情報を知らない**
  - **局所的な情報で決定**する
  - ノードの**故障**はアルゴリズムに**影響しない**
  - **絶対時計(グローバルクロック)は仮定しない**

スケーラビリティの問題（2）

- 地理的なスケーラビリティの問題
  - LANでは同期通信で問題ない（～数百us）がWANでは大問題（～数百ms）
  - LANは高信頼，マルチキャストも可だがWANはパケットロスがあり，一対一が基本
    - サービス発見はLANだとブロードキャストすればよいが，広域だと地球規模，数十億人規模でスケールする設計が必要
  - 集中型のシステムはWANの遅延，低信頼性により資源の有効利用ができない

スケーラビリティの問題(3)

- 複数の管理ドメインをまたがる分散システム
  - 資源の利用法，管理，セキュリティなどポリシの違いを吸収する必要がある
  - 通常，組織の管理者は信頼するが，他の組織の管理者を同様に信頼できるか？
  - 他の組織のユーザに対して異なる権限，アクセス制御が必要？
  - 他組織の信頼できないプログラムの実行

スケールするための技術

- **通信遅延隠蔽**
  - 遠隔のサービスの返答を待たない＝**非同期通信**
  - 通信量を減らす
    - クライアントにサーバ側処理（入力チェックなど）を移動
- **分散**
  - 要素を分割して分散させる
  - DNSの例：階層的なドメイン名を重複のないゾーンに分割。それぞれのゾーンは単一サーバで処理
  - WWWの例：文書を分散格納しているからスケールする
- **複製**
  - 可用性の向上，負荷分散，通信遅延隠蔽
  - **キャッシュ**：通常オンデマンドでクライアント主導
  - 複製間の**一貫性**制御：強さは利用形態（Web vs auction）による

落とし穴

- 起こしやすい間違った仮定
  - ネットワークは高信頼である
  - ネットワークはセキュアである
  - ネットワークは均一である
  - ネットワークトポロジは変化しない
  - ネットワーク遅延はない
  - ネットワークバンド幅は無限である
  - 通信のコストはない
  - 管理者は一人

分散システムの型

- 分散コンピューティングシステム
  - クラスタコンピューティングシステム
  - グリッドコンピューティングシステム
- 分散情報システム
  - トランザクションプロセッシングシステム
  - エンタープライズアプリケーション統合(EAI)
- 分散ユビキタスシステム

分散コンピューティングシステム

- ハイパフォーマンスコンピューティング
- クラスタコンピューティング
  - 高速ネットワークで接続されたPC
  - 比較的均質な環境（計算機，OS，ネットワーク）
- グリッドコンピューティング
  - 複数の組織の資源を仮想組織により利用
  - スパコン，クラスタ，ストレージ，データベース，実験装置，センサ
  - 非均質な環境（管理ドメイン，セキュリティ）

分散情報システム

- 複数のサーバ（データベース）をクライアントが利用
  - ホテルと航空券の予約
- ネストトランザクション
  - 複数のサブトランザクションで構成
  - どれかのサブトランザクションが失敗したら全体が失敗

分散ユビキタスシステム

- モバイルや組込デバイス，ワイヤレス
- 環境の変化に適応
- 様々なアクセス方法に適用
- 簡単にアクセス可能でなければならない

まとめ

- 分散システムは，複数の**自立**した計算機を**単一**の**一貫**したシステムとして**みせる**
  - 複数計算機で実行されているアプリケーションを統合
  - 設計次第ではネットワークのサイズに応じてスケール可能
  - ソフトウェアの複雑さ，性能の低下，セキュリティの問題も考慮する必要がある
- **分散透明性**の実現は分散システムの目標であるが
  - 透明性の程度と性能は**トレードオフ**の関係
- ネットワークの間違った仮定
- さまざまな分散システムの型

演習問題

- 分散システムの定義は何か？
- 分散透明性とは何か？
- 分散透明性の程度とシステムの性能，効率的な利用について論じよ
- サイズや距離に関してスケールするための技術にはどのようなものがあるか？