

# 分散システム 第1回

## — 分散システム概説 —

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

# 担当者

---

■ 大森 隆行

おおもり たかゆき

OMORI Takayuki

# 授業の目的

---

- 分散システム(distributed system)に関する基礎知識の習得
  - 分散システムを構築するために必要な概念やアルゴリズムについて説明できる
  - また、それらを分散システムの構築にどのように適用できるかについて説明できる
  - 分散システムの実例を取り上げ、その機能や構成について説明できる

# この授業について

---

## ■ 15回の講義

- 15回の講義とは別に期末試験があります

## ■ 講義の流れ

- 説明 → 確認問題 → 確認問題の説明

## ■ 評価方法

- 日常点（出席、レポート、中間試験） 40%
- 期末試験 60%

# 授業計画

---

- 第1回 分散システム  
概説
- 第2回 アーキテクチャ
- 第3回 プロセス
- 第4-5回 通信
- 第6回 名前
- 第7回 同期
- 第8回 一貫性と複製
- 第9回 中間試験
- 第10-11回  
フォールトトレラント性
- 第12回 セキュリティ
- 第13-14回 Web  
アプリケーション
- 第15回 まとめ

※内容や理解の程度によって多少前後することがあります

# 参考書

---

- 「分散システム」  
水野 忠則 監修、共立出版、2015
- 「分散システム 原理とパラダイム 第2版」  
アンドリュー・S・タネンバウム 他 著、2009

# その他

---

- スライドはWebサイトにアップロードします  
<http://www.ritsumei.ac.jp/~tomori/ds/>
  - あらかじめダウンロードして、講義中に手元で参照できるようにしておくことをお勧めします
  - 講義の数日前にはアップロードします
  - 修正があった場合は再アップロードします  
(更新日は上記サイトに掲載)

# 講義内容

---

## ■ 分散システム概説

- 分散システムとは
- 分散システムの目的
- 分散システムの例
- 分散システムの種類
- 分散システムの設計目標



# 分散システム

---

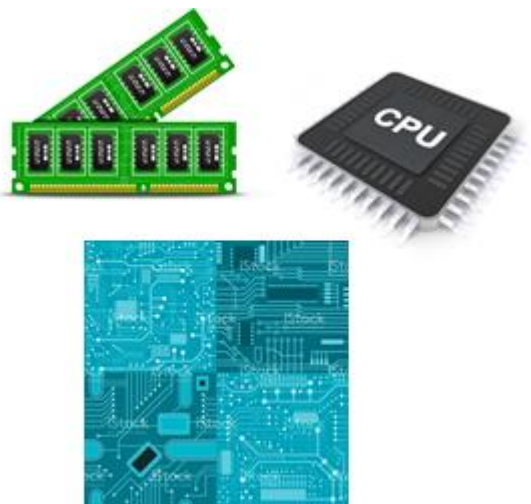
- 分散システムはそのユーザに対して  
単一的首尾一貫した (coherent) システム  
として見える  
独立したコンピュータの集合である
- A distributed system is  
a collection of independent computers  
that appears to its users  
as a single coherent system.

\* A.S. Tanenbaum et al.

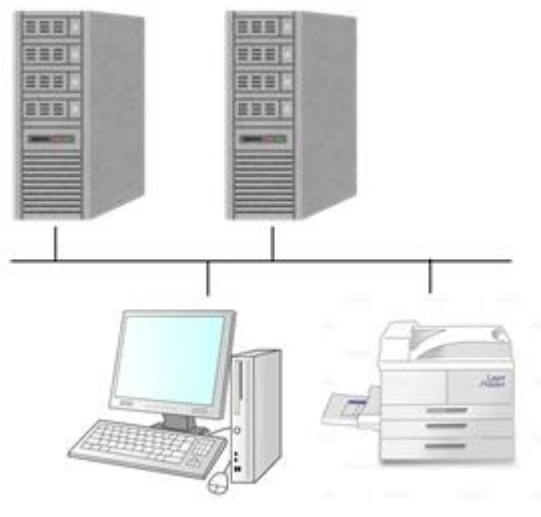
"Distributed Systems - Principles and Paradigms -"

# 分散システム

- 独立したコンピュータの集合  
→ 単体でも自律している(autonomous)  
コンピュータの集合



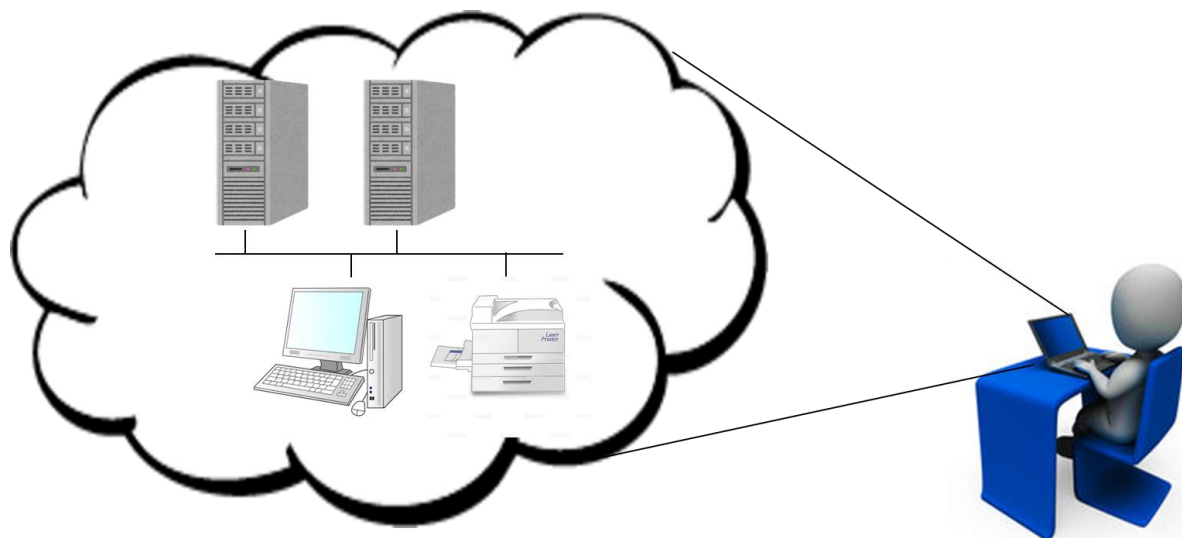
自律していないものの  
集まり (= 部品)



自律したものの集まり

# 分散システム

## ■ 単一の首尾一貫した (coherent) システム



- 複数のコンピュータで構成されていても  
利用者はその詳細は知らない(知る必要がない)  
→ 個々のコンピュータは協調しなければならない

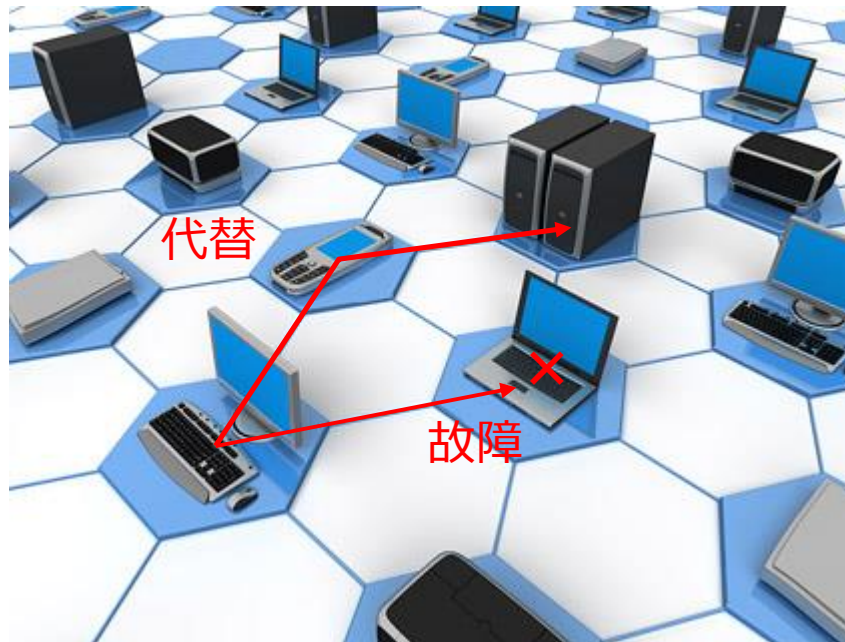
# 分散システムの目的

---

- 一台のコンピュータでは難しいことを可能にする
  - 可用性 (availability)
  - リソースへのアクセス可能性向上
  - 局所性 (locality)
  - スケーラビリティ (scalability)

# 分散システムの目的 - 可用性

- システムのサービスを利用したいときに利用できる
- 分散システムでは、1台が故障してもサービスの継続が可能



# 分散システムの目的 - アクセス可能性向上

---

- ユーザ(またはアプリケーション)が遠隔地にある資源(リソース)へのアクセスをすることを可能にする
  - リソースの例：プリンタ、コンピュータ、ストレージ、装置、データ、ファイル、Webページ、ネットワーク
- 経済性：個別にリソースを保持するより安価
- 情報交換のしやすさ
  - グループウェア(groupware)による共同作業等

# 分散システムの目的 - 局所性

---

## ■ 保護すべき情報を局所的に保存

## ■ 例：電子メール

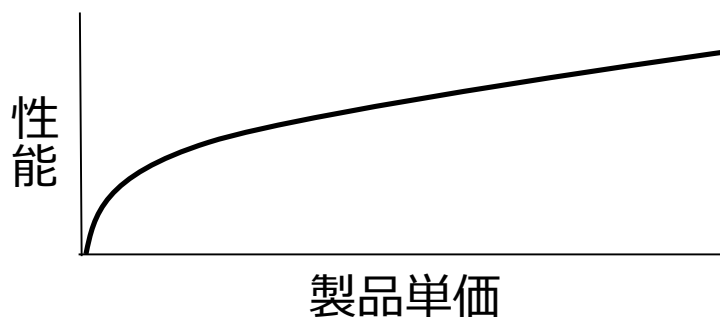
- 学内宛のメールは学内サーバで対応
- 学外宛のメールは学外のサーバへ転送
- メール処理をするサーバが1箇所に集中していると、宛先に関わらずそのサーバへ送ることになる

# 分散システムの目的 - スケーラビリティ

- コンピュータの数を増やすことで変化に柔軟に対応

- 集中システムの問題

- ユーザ数が10倍に増えたら？



10倍の性能を得るためのコストは  
10倍では済まない

- 稼働中に規模を拡大したくなったら？

- 設計段階で規模拡大を想定しておく必要



# 確認問題

- ()内から正しい語句を選択せよ。
  - 分散システムはそのユーザに対して(単一・複数)の首尾一貫したシステムとして見える独立したコンピュータの集合である
- 以下は分散システムの性質について説明したものである。それぞれ何という性質か。語群から選択せよ。
  - システムのサービスを利用したいときに利用できるという性質
  - ユーザが遠くのリソースにアクセスできるという性質
  - システムの規模を拡大しやすいという性質

語群： スケーラビリティ・局所性・アクセス可能性・可用性
- 次の文は正しいか。○か×で答えよ。
  - 分散システムでは、利用者はシステムが多くのコンピュータで構成されていることを意識しなくても良い



# 分散システムの例

---

- WWW、電子メール
- 銀行のオンラインシステム
- クラウドアプリケーション

等々 . . .

# 分散システムの種類

---

- 分散コンピューティングシステム
- 分散情報システム
- 分散パーベイシブシステム

# 分散コンピューティングシステム

---

## ■ 目的：

高性能計算、信頼性の高いデータ保管等

## ■ クラスタコンピューティング cluster computing

- 複数のPCやワークステーションで構成

- 高速LANで結合

## ■ グリッドコンピューティング grid computing

- ネットワーク上に分散した  
複数のサブグループにより構成

- 複数の管理組織にまたがる

# 分散情報システム

---

- 大学・企業等による  
ネットワークアプリケーションの提供
- トランザクション処理システム
  - 遠隔からのデータベース操作
  - 操作はトランザクション単位で実行
- エンタプライズアプリケーション統合  
(EAI: enterprise application integration)
  - 企業内で使用される複数の分散アプリケーションを統合
  - 遠隔手続き呼び出し(RPC)や  
遠隔メソッド呼び出し(RMI)等を利用

# ACID特性

---

オールオアナッシング特性(all-or-nothing property)とも呼ぶ

- 原子性 (Atomic)

- トランザクションは外部から見て分割不可

- 一貫性 (Consistent)

- トランザクションはシステム不変性を侵蝕しない

- 独立性 (Isolated)

- 並行トランザクションは互いに干渉しない

- 永続性 (Durable)

- トランザクションがコミットするとその変更は永久

# 分散パーベイシブシステム

---

- pervasive: 遍在する
- ユビキタスシステム(ubiquitous system)ともいう
- 小型のセンサやその他の機器を互いに連結
- 要素技術
  - 無線アクセス (無線LAN、Bluetooth等)
  - RFID(radio frequency identification)
    - 無線認証を用いて物体の認証を行う
  - センサネットワーク
  - 位置特定技術 (GPS等)



# 確認問題

---

- 次の説明はクラスタコンピューティング、グリッドコンピューティングのどちらに該当するか。
  - LANで各マシンが結合される
  - ネットワーク上に分散した複数のサブグループにより構成される
- 以下の各文はトランザクションのACID特性について説明したものである。各説明はどの特性に関するものか。
  - 並行するトランザクションは互いに干渉しない
  - 1つのトランザクションは外から見て分割できない
  - トランザクションはシステム不変性を侵さない
  - コミットされたトランザクションによる変更は永久的である



# 分散システムの設計目標

---

- 分散透過性 (distribution transparency)
- 開放性 (openness)

# 分散システムの設計目標 - 分散透過性

- 分散されている事実をユーザ(やアプリケーション)から見えなくすること

透過性の種類 [ISO, 1995]

アクセス	データ表現における相違、リソースがどのようにアクセスされているかを隠蔽
位置	リソースの位置を隠蔽
移動	リソースが他の位置に移動して良いことを隠蔽
再配置	リソースが使用中に他の位置に移動するかもしれないことを隠蔽
複製	リソースが複製されていることを隠蔽
並行	リソースがいくつかの競合ユーザに共有されていることを隠蔽
障害	リソースの障害と回復を隠蔽

# アクセス透過性

---

- 異なるマシンやOSによるアクセス方法 (e.g., データ表現の違い) を隠蔽
- (例) ファイルの名前付け規約の違い
  - C:¥Users¥omori¥Document¥Book.pdf (Windows)
  - Macintosh HD:Users:omori:Document:Book.pdf (Apple HFS)
  - /Users/omori/Document/Book.pdf (UNIX)

# 位置に関する透過性

---

- 位置透過性 (location transparency)
  - システム内のどこにリソースが物理的に存在しているかを隠蔽する
  - (例)<http://www.ritsumei.ac.jp/>では, 立命館のサーバがどこにあるのかを意識させない
- 移動透過性 (migration transparency)
  - リソースを移動しても同じアクセス方法を使用可能
- 再配置透過性 (relocation transparency)
  - 利用中の移動が可能
  - (例)無線パソコンをネットワークに接続したまま移動できる

# システム構成に関する透過性

---

- 複製透過性 (replication transparency)
  - 可用性や性能を向上させるために配置するコピーを意識させない
- 並行透過性 (concurrency transparency)
  - 共有リソースへの同時アクセスに対して一貫性を保証する
- 障害透過性 (failure transparency)
  - リソースの障害と回復を意識させない

# 透過性の度合い

---

- 完全な透過性を提供することは技術的に不可能
  - (例) San FranciscoのプロセスとAmsterdamのプロセスの間で35ms以下で通信したい → 不可能
- 完全な透過性を提供すれば良いとも限らない
  - (例) 一部のサーバで障害  
→ 隠蔽しようとする则全体のサービスの速度が低下



# 分散システムの設計目標 - 開放性

---

- プロトコル(protocol)が開かれていること
  - プロトコルに従った機器を誰でも開発可能
  - プロトコルに従った機器は相互運用可能(interoperable)
  - プロトコルに従った機器を別のシステムで利用可能(portable)
- プロトコルが開かれていることにより、機器の置き換えや拡張が容易になる

# 陥りやすい誤解

---

- ネットワークは信頼性がある
- 遅延は存在しない
- 帯域幅は無限である
- ネットワークはセキュアである
- トポロジは変化しない
- 存在する管理者は一人である
- 転送コストはゼロである
- ネットワークは均一である

# 確認問題

- ()内から正しいものを選べ。
  - 一般的に、個々のコンピュータの性能を10倍に上げるためのコストは10倍より(高い・低い)
- 次の説明に合う用語を答えよ。
  - システムの構成要素が実際には分散されている事実をユーザから見えなくすること
  - 誰でも参照できるようにプロトコルが公開されていること
- 以下はどの透過性に関する説明か。
  - マシンごとのリソースへのアクセスの方法の違いを意識させない
  - リソースの障害と回復を意識させない
  - リソースを移動してもアクセス方法は変えなくて良い
- 分散システムにおけるプロトコルが開放されていることの利点を3つ挙げよ。



# 参考文献

---

- 「分散システム」  
水野 忠則 監修、共立出版、2015
- 「分散システム 原理とパラダイム 第2版」  
アンドリュー・S・タネンバウム 他 著、2009
- 「はじめての分散処理システム」  
菅原研次著、森北出版、2007