クラウドシステム基礎

第１回（前）：イントロダクション

国立情報学研究所
石川　冬樹
f-ishikawa@nii.ac.jp

本講義の位置づけ

- 分散システムの難しさ（重要さは言わずもがな）
  - 相互運用性からセキュリティまで，多様な側面
  - それらの間のトレードオフ（特に，複製管理における性能，一貫性，および耐故障性・可用性）
- これまで累積された知見の活用に向けて
  - 基礎知識の習得
    - 様々な種類の「一貫性」など，達成する性質の，厳密な定義および実現のための原則や方法
  - クラウドにおけるそれらの役割や活用の議論

教科書・参考書(１)

- 分散システム　第二版
  - A. Tanenbaum and M. Steen(水野ら訳)
    ピアソン桐原, 2009
  - 700ページ強
  - 一部を参照(「初めに」と,「同期」,「一貫性と複製」,「フォールトトレラント性」)
  - 広く概観できる, いわゆる「教科書」

教科書・参考書(２)

- Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services
  - K. Birman, Springer-Verlag New York, 2012
  - 700ページ強
  - 一部を参照(Introductionと，Part IIにおける複製管理などの手法)
  - 前述の教科書よりも，理論と実践のギャップや，クラウドに関する背景や利用シナリオ，技術などへの言及があり，意義や位置づけが参考になる

分散システム：定義

- 「分散システムは，ユーザに対して単一で首尾一貫（coherent）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」

  教科書「分散システム 第2版」より

- その重要性は言わずもがな
  - 個々のコンピュータの高性能化・低価格化による発展
  - 様々な基盤技術・先進技術に含まれる特性（WWW，クラスタ，オンラインゲーム，クラウド，大規模分散処理，センサーネットワーク，・・・）

分散システム：目標（１）

- ユーザとリソースの容易な（セキュアな）接続
  - 小さく安価なものを多く用いる経済性・拡張性
  - 物理的に離れた地点間でのリソース活用や情報共有
- 透過性
  - 分散に起因する複雑さを隠蔽
  - 例：アクセス透過性
    （データ表現やアクセス方法の違いを隠蔽）
  - アクセス，位置，移動，再配置，複製，並行，障害
    [ISO 1995]

分散システム：目標（２）

- オープン性
  - 十分な情報を含む一方で「どう作るか」を規定しないインターフェース定義に基づいた相互運用性
  - （他の部品に影響を与えず）異なる開発者による部品の追加・入れ替えによる柔軟性・伸張性
- スケーラビリティ
  - より多くのユーザやリソースに対する実現性・性能
  - 距離的に遠隔な，広い範囲での拡張性
  - 独立した複数組織にまたがっての管理性

分散システム：話題（１）

- 教科書「分散システム 第2版」の目次
  - アーキテクチャ（集中型と非集中型，自己管理のためのモデルなど）
  - プロセス（スレッド，仮想化，クライアントとサーバ，コード移動など）
  - 通信（RPC，メッセージ，ストリーム，マルチキャストなど）
  - 名前付け（名前空間，ディレクトリサービスなど）

分散システム：話題（２）

- 教科書「分散システム 第2版」の目次（続）
  - 同期（論理クロック，相互排他，選任など）
  - 一貫性と複製（一貫性モデル，レプリカ管理，一貫性プロトコルなど）
  - フォールトトレラント性（プロセスの回復力，高信頼通信，分散コミット，回復など）
  - セキュリティ（セキュアチャネル，アクセス制御，セキュリティ管理など）

分散システム：話題（３）

- 教科書「分散システム 第2版」の目次（続）
  - 分散オブジェクト指向システム
  - 分散ファイルシステム
  - 分散ウェブベースシステム
  - 分散協調ベースシステム

本講義における焦点(１)

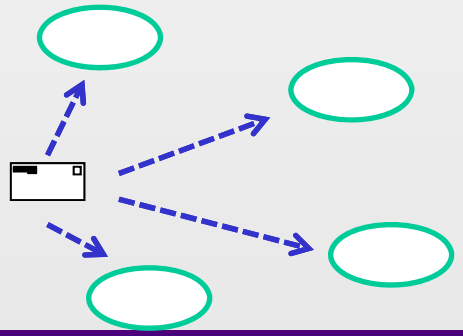
- 同期や一貫性，耐故障性に焦点を当てる
  - 複雑であり，本質的な限界やトレードオフもあるため，難しい側面
  - クラウドコースの焦点である「大規模化」には必須の側面
    - 既存クラウドサービスの設計指針を理解し，適切な活用方法を議論する上でも重要
  - 現場で，「とにかく動かす」，ようにやっていると，一般論を習得，議論したり，じっくりと分析や議論をする機会がなさそう(？)

本講義における焦点(２)

- その他の観点は，概観する程度
  - 「動かす」ための技術は皆さん詳しい
    - 通信機構やRPC，名前空間など実用性の高い基礎概念やそれらを扱う最新のフレームワーク・ミドルウェア
  - セキュリティについては，より広い視点からの講義もあり(概論，要求分析)
  - 歴史的な分散OS，ミドルウェアなどを学ぶことはかなりより勉強になるが，文献も多いので割愛(ぜひ教科書1個目の該当部分を)

典型的な問題設定：状況

- ソフトウェア部品や仮想マシンを複製し，複数の物理サーバに配置
  （例：Webショッピングサイトにおけるビュー処理や分析処理，データ管理処理などの一機能）
  - （非常に）大量のリクエストに対応するため
  - 部分的なノードの故障に耐えるようにするため
- 同一メッセージをマルチキャストすることも多い



例：
多数のクライアントによるリクエストをノードに分散．
情報の追加・更新・削除など，
処理の一部は全ノードで共有される．
（処理のコマンドまたは結果をマルチキャスト）

典型的な問題設定：懸念事項

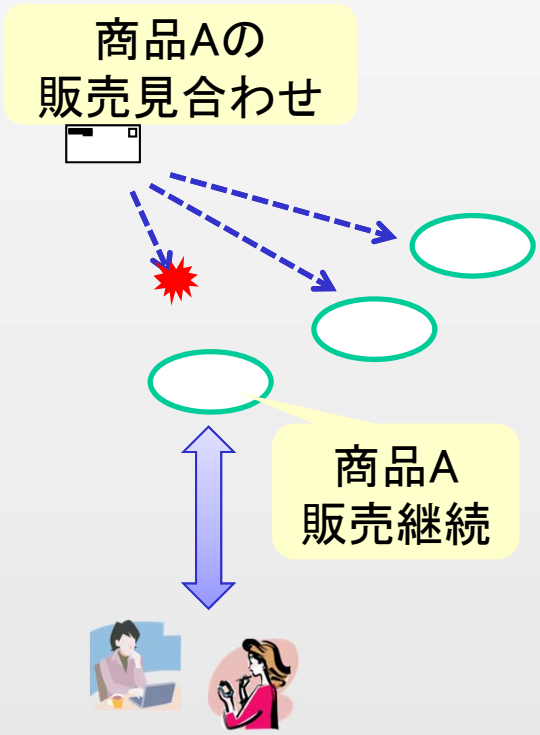
- 信頼性をゆるがしうる多数の要因

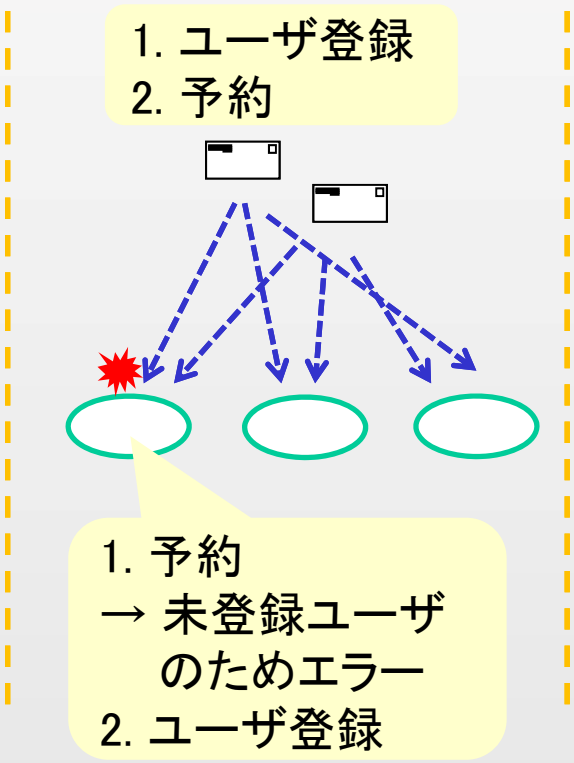
  機械の故障，機械やソフトウェアの高負荷による挙動変化，バグ，運用操作のミスなどにより，

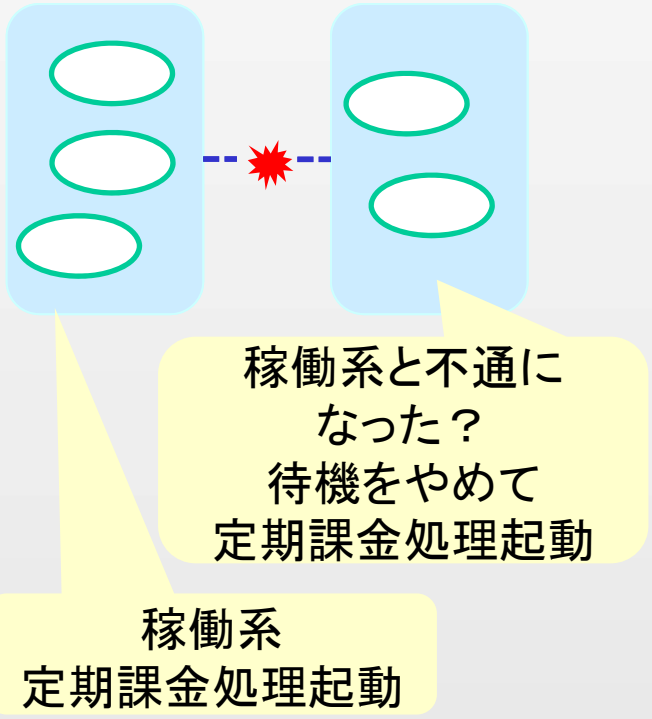
  - 一部のノードがクラッシュする（そして再起動などを経て復帰する）
  - ノードが追加されたり削除されたりする
  - ネットワークが複数の領域に分断される
  - 一部のノードへのマルチキャスト配信が失敗する
  - 複数のマルチキャスト配信が，異なる順序で様々なノードに届く

典型的な問題設定：考えられる不具合

- あくまでアプリケーション上の要求に依存するが，下記は常識的に問題になりそうな状況の一例







典型的な問題設定：保証したい性質

- 保証したい性質の例
  （できればフレームワーク・ミドルウェア側で）
  - ノードのグループすべてがある更新を反映するか，どのノードも反映しないか，いずれかにしたい
  - メッセージのノードへの到着順序が変わっても，アプリケーションには，ある特定の「正しい」順序で配達される（どんな順序？）
  - 自分たちが処理の主導権を握っている・責任を負っていると判断しているノードのグループが，ある時間に二つ以上存在することはない

典型的な問題設定：実現の難しさ

- ノード間において，今後の処理の進め方または処理が進んだ結果について確認し，合意を得るため，何かしら追加の情報交換を行うことになる
  - システム全体として起きうる状態遷移がややこしく，理解，設計，検証が難しい
  - 達成できる性質と，その達成のためのオーバーヘッド（実行時間）にトレードオフがある
  - 通信発生や故障発生の頻度や傾向に依存して，実現方法を調節し最適化できたり，逆に特定の実現方法の性能が悪くなってしまったりする

最近の状況

- クラウドでは，スケーラビリティ・性能と可用性を重視し，一貫性（整合性）や検索機能などに関し制限があると言われる
  - 設計思想が異なる
  - 利用する際の留意事項，特徴や留意事項を説明する言葉も異なる

最近の状況(続)

- 概念(用語)を正確に理解，議論できている？
  - 例：「BASEトランザクションをサポートしている」
  - 例：「CAP定理を踏まえると・・・」
  - 例： Amazon Dynamo DBのドキュメントより

    「GetItemは結果整合性のある読み込み」

    「アトミックカウンターがサポートされています」

    「条件付き書き込みはべき等のオペレーション」

    http://docs.aws.amazon.com/ja_jp/amazondynamodb/latest/developerguide/WorkingWithItems.html

本講義(担当講師)のスタンス

- 「基礎(理論や技術)」を学んでみましょう
  - 既存の知見(特に設計)を再利用して解決できるに越したことはない
  - 技術者として，正確な定義や分析，議論ができるよう体験すべき(＋ 常識として知っておくべき？)
- その実用的な活用是非や方法を議論しましょう
  - たいてい，現場にはオーバースペック？
  - ときどき，重要視する観点が現場と異なる？

(が，「良さ」の基準や理想のあり方を示すよい道しるべになる，よいスタート地点・たたき台になる)

講義スケジュール

- 全7コマ
  - 1: イントロダクション，実現基盤概観
  - 2 – 4: 同期や一貫性，耐故障性に関する
    基礎知識・技術
    （ある程度古典的・教科書的）
  - 5 – 7: 分散システムとしてのクラウド
    （現状や実用とよりからめて）

2回目，6回目の内容についてグループ演習

評価

- 出席（講義内にめいっぱい考えましょう）
  - 欠席の場合には，講義内容や講義内演習に関するレポートまたは講師とのやりとり
- レポート課題1つ
  - 今日の4コマ目まで終わればできる
  - 「特定の知識が要らない，比較的簡単なものをじっくり考える機会をとってみよう」

    なので，今すぐでもおそらくできる