

調査 5 モデルカリキュラムの提言 コースウェア

4. 分散アーキテクチャに関するスキル

I. 概要	分散情報システムを構築する際によく用いられる要素技術とその基礎となる考え方を理解するとともに、実際のミドルウェアやアプリケーションの事例に即した形で、分散アプリケーション構築に必要なスキルを身につける。
II. 対象専門分野	職種共通
III. 受講対象者、 受講前提	<ul style="list-style-type: none">• Web やメールなどの基本的なインターネットアプリケーションの利用経験があること• ファイル、ディレクトリ、プロセスなどの OS の基本概念を理解していること• 確率に関する初歩的な知識があること• 複数のプログラミング言語を使った経験があること
IV. 学習目標	<ul style="list-style-type: none">• 分散アーキテクチャの重要性と特徴を理解する。• 名前管理、複製管理、多重化、セキュリティ、遠隔呼び出しなどの要素技術の重要性、必要性、さらに基本的な仕組みを理解する。• 実際のインターネットアプリケーションで分散アーキテクチャの要素技術がどのように利用されているかを学ぶ。• CORBA、Web サービスなどの基本的な仕組みとこれらを用いた分散アプリケーション構築の手順を学ぶ。• 既存の Web API を用いたマッシュアップ型のアプリケーション構築の手順を学ぶ。• 分散トランザクション、モビリティ、ピアツーピアアーキテクチャなどの高度な要素技術の基本となる考え方を理解する。

<p>V. 使用教科書、 教材等</p>	<p>この分野を網羅的にカバーした日本語の書籍は稀であり、以下の訳書がほぼ唯一のものであろう。これは、800 ページを超える大著であり、入門的内容から高度な内容まで含まれている。レベル的には、入門的な内容がレベル 1、高度な内容がレベル 3 に対応する。レベル 2 はプログラミング演習が主となり、演習環境に大きく依存するため、オリジナル教材を用いる。</p> <p>『分散システム—原理とパラダイム』 アンドリュー S. タネンバウム、マールテン ファン スティーン著 水野忠則、鈴木健二、佐藤文明、宮西洋太郎、西山智、東野輝夫訳 ピアソンエデュケーション刊</p>
<p>VI. 習得スキル の 評価方法</p>	<p>カリキュラムの内容に関する各種手法の利害得失等を問う記述式の試験問題 5 問に解答し、80%の正答率をもって合格とする。</p>
<p>VII. カリキュラム の 構成</p>	<p>レベル 2 第 1 回～第 15 回</p>

講座内容

第1回 分散アーキテクチャとは(講義 90分)

コンピュータとネットワークの発展史を概観し、分散情報システムが登場した理由を理解する。

そして、分散アーキテクチャのメリット、分散情報システムに求められるもの、実際の分散アプリケーションで採用されているアーキテクチャの事例を学ぶ。安価な機器が大量にネットワーク接続される時代に適した情報システムの構成法としての分散アーキテクチャの意義を理解する。

(1) 集中と分散の歴史

1. メインフレーム、ミニコンの時代
2. PC と LAN の時代
3. Web とインターネットの時代

(2) 分散情報システムの特徴と留意点

1. 耐故障
2. スケーラビリティ
3. 地理的分散
4. 分散と共有の両立
5. 相互運用性

(3) さまざまな分散アーキテクチャ

1. 計算の分散化(クライアントサーバ、マスタワーカ)
2. ストレージの分散化(SAN、分散ファイルシステム、データベースシステム、P2P ファイル共有)
3. ユーザインタフェースの分散化(シンクライアント、リッチクライアント、Ajax)
4. 情報システム全体の分散化(Nティアモデル)
5. 統合的なビジョン(オートノミックコンピューティング)

第2回 名前管理(講義+ワークショップ 120分)

分散情報システムでは、資源にアクセスするために論理的な名前を用いることが多い。この名前という考え方の重要性、名前を介したアクセスが一般のインターネットアプリケーションでも広く使われていること、名前と密接に関連した透過性という概念と分散アーキテクチャの関係について理解する。最後に、LDAP、DNSなどの実際に使われている分散名前管理システムの大まかな仕組みを理解し、コマンドを使ってこれらにアクセスする方法を学ぶ。

(1) 名前を介したアクセス

1. 論理的でユニークな名前の必要性
2. 階層的な名前空間
3. コンピュータの名前としてのドメイン名
4. Web ページの名前としての URL
5. ファイルの名前としてのパス名

(2) 名前と透過性

1. 透過性とは
2. さまざまな透過性(アクセス透過性、位置透過性、複製透過性など)
3. シングルシステムイメージ
4. 透過性はどこまで必要か?

(3) 分散名前管理システムの事例

1. ディレクトリサービス(LDAP)
2. 広域分散名前サービス(DNS)

(4) 分散名前管理システムに対する検索(ワークショップ)

1. ldapsearch、dig 等の使い方

第3回 複製管理(講義+ワークショップ 120分)

分散情報システムが管理するデータの可用性とアクセス性能を向上させる技術としての複製管理の重要性を理解し、その概要を学ぶ。さらに、複製の存在がシステムの性能に与える影響を推定するために重要となるキャッシュのヒット率を概算する方法を学ぶ。

(1) 複製のメリット

1. データ消失の防止
2. アクセス性能の向上(スループット、遅延)

(2) さまざまな複製技術

1. ファイルのバックアップと同期
2. ディスクキャッシュ
3. RAID 0/1/5
4. Web プロキシ

(3) 複製間の一貫性管理

1. 一貫性の問題
2. 単一マスターコピー
3. コヒレンシプロトコル

(4) キャッシュのヒット率のモデル(ワークショップ)

第4回 耐故障性(講義+ワークショップ 120分)

情報システムが社会基盤化し、組織の基幹部門での利用も進んだ結果、システム障害が大きな被害を招く事例が増えてきた。一方で、ハードウェアの故障やソフトウェアのバグを完全になくすことは困難である。不安定な機器やソフトウェアから構成される情報システムを安定して稼働させることの重要性を理解し、そのための基礎となる技法を学ぶ。そして、分散情報システムでは、冗長性を持たせた機器構成が一般的であり、うまく設計、運用すれば耐故障性や可用性を高めやすいことを理解する。最後に故障確率を数学的に議論する方法の概要を学ぶ。

(1) 分散システムと故障

1. 情報システムの障害事例とその社会的影響
2. 故障(fault)、誤り(error)、障害(failure)
3. 単一点障害

(2) コンピュータの冗長化と故障からの回復

1. ホットスタンバイ
2. 投票と合意
3. チェックポイントとリカバリ

(3) ネットワークの冗長化

1. レイヤ2とレイヤ3
2. レイヤ2での対応(RSTP、MST)
3. IP経路制御による迂回(RIP、OSPF)
4. ルータのホットスタンバイ(VRRP、HSRP)

(4) 不安定な通信路上の安定した通信

1. TCPの仕組み
2. メール配送の仕組み

(5) ソフトウェアの冗長化

1. Nバージョンプログラミング

(6) 故障の確率モデル(ワークショップ)

第5回 セキュアチャネル(講義 90分)

分散アプリケーションの中には、インターネットのようなセキュアとは限らないネットワークを利用するものが少なくない。このようなネットワーク上にセキュアな通信路を論理的に構成する方式の重要性を学び、暗号技術を利用することでそれが実現可能であることを理解する。また、セキュアな通信路の事例としてSSL/TLSの基本的な仕組みを学ぶ。

(1) セキュアチャネルの特徴と実現技術

1. 機密性、完全性、認証
2. 公開鍵暗号系(RSA)
3. セキュアハッシュ関数
4. 電子署名

(2) SSL/TLSの仕組み

1. PKI
2. 交渉、認証、鍵交換
3. MAC

第 6 回 CORBA 入門(1)(講義+ワークショップ 120 分)

分散アプリケーション構築基盤としての CORBA の概要を理解し、CORBA によるアプリケーション開発の手順を学ぶ。同時に、遠隔呼び出しに必要なコードの大部分が自動生成可能なことを学ぶ。

(1) CORBA の概要

1. 遠隔メソッド呼び出し
2. スタブとスケルトン
3. IDL
4. ORB
5. 相互運用性(GIOP、IIOP)

(2) CORBA を用いたプログラムの開発演習(ワークショップ)

1. IDL コンパイラ
2. サーバ側、クライアント側のプログラム作成
3. ディプロイメント
4. ORB とアプリケーションの起動

第 7 回 CORBA 入門(2)(講義+ワークショップ 120 分)

CORBA が提供するさまざまな機能を概観するとともに、非同期メソッド呼び出し等を含むアプリケーション開発の手順を学ぶ。

(1) CORBA が提供する機能

1. 名前サービス
2. 永続オブジェクト
3. トランザクション
4. セキュリティ

(2) CORBA を用いたプログラムの開発演習(ワークショップ)

1. 非同期呼び出しとコールバック
2. オブジェクトの参照
3. 複数言語による分散アプリケーション開発

第 8 回 Web サービス入門(1)(講義+ワークショップ 120 分)

サービスとしてのアプリケーションという考え方と Web サービスが登場した背景を学ぶ。
また、既存の Web サービスを利用するクライアント側プログラムの開発手順を学ぶ。

(1) Web Services の概要

1. サービスとは？
2. ファイアウォールと相互運用性
3. WSDL
4. XML
5. SOAP

(2) WSDL からのクライアント側プログラムの開発演習(ワークショップ)

1. WSDL ファイルの取得
2. スタブの生成
3. クライアント側プログラムの作成

第 9 回 Web サービス入門(2)(講義+ワークショップ 120 分)

Web サービスに関するさまざまな規格を概観するとともに、サーバ側アプリケーション開発の手順を学ぶ。

(1) Web Services に関係したさまざまな規格

1. UDDI
2. BPEL
3. WS-Transaction
4. WS-Security

(2) WSDL を用いるサーバ側プログラムの開発演習(ワークショップ)

1. IDL、インタフェース等からの WSDL の生成
2. サーバ側プログラムの作成

第 10 回 マッシュアップ入門(講義+ワークショップ 180 分)

大手ポータルサイトをはじめとして、Web アプリケーションの API を公開する事例が増えている。これらの API を活用することで、比較的簡単に高機能なアプリケーションを実現できることを学ぶ。ワークショップでは、JavaScript を用いて Web API を操作する演習を行う。

(1) マッシュアップとは

1. マッシュアップの事例 (Programmable Web など)
2. Web API
3. 主要なメッセージ形式とプロトコル (REST、JSON、RSS、ATOM など)

(2) Web API を用いたマッシュアップの演習 (ワークショップ)

1. Web API の事例紹介
2. JavaScript 入門
3. HTML と JavaScript を用いたアプリケーションの開発

第 11 回 分散トランザクション(講義 90 分)

機器の故障から重要なデータを守り、さらに並行性制御の難しさを緩和するために利用されるトランザクション技術の概要を理解し、その分散版である分散トランザクションの要素技術を学ぶ。また、長時間トランザクションの必要性とそれを実現するための妥協策を理解する。

(1) トランザクションとは？

1. 不可分な一連のアクションの必要性
2. ACID (Atomicity、Consistency、Isolation、Durability)

(2) 分散トランザクション

1. 合意の必要性
2. コミット問題
3. 2フェーズコミット (3フェーズコミットはオプション)

(3) 長時間トランザクション

1. 長時間トランザクションの必要性
2. 補償トランザクション

第12回 ピアツーピアアーキテクチャ(講義 90分)

ピアツーピア(P2P)は、情報漏洩と関連づけて論じられることが多く、悪いイメージが先行した感がある。一方、ピアツーピア型の分散アーキテクチャをうまく利用することで、スケーラビリティと耐故障性を高いレベルで実現したシステムも少なくない。ピアツーピア型アーキテクチャの特徴を理解し、実際の活用事例について学ぶ。

(1) ピアツーピアの特徴

1. 耐故障性とスケーラビリティ
2. 動的なピアの参加と離脱
3. 匿名性との関係
4. 管理コスト
5. 法律問題

(2) ルックアップの方式

1. 中央サーバ方式
2. ブロードキャスト方式
3. 分散ハッシュテーブル

(3) ピアツーピアの事例

1. ファイル共有とコンテンツ配信(BitTorrent、Groove など)
2. コミュニケーション(Skype、Jabber など)

第 13 回 モビリティ(講義 90 分)

今日の情報システムは、携帯電話、PDA、ノート PC などの移動する機器を構成要素として含むことが多くなった。一方、Java アプレット、JavaScript、Active X コントロールのようにソフトウェアの移動も頻繁に発生している。機器やソフトウェアの移動により享受できる利便性と解決すべき問題について理解するとともに、円滑な移動を支援するために開発された要素技術について学ぶ。

(1) 移動の利点と欠点

1. 利便性
2. 通信性能への影響
3. セキュリティドメインを超える移動

(2) 機器の移動

1. 構成変更とサービスの発見 (DHCP、UPnP、Jini など)
2. モバイル IP
3. アドホックネットワーク
4. Disconnected Operations (Coda ファイルシステム、Google Gears)

(3) コードの移動

1. 互換性
2. サンドボックス
3. ライブマイグレーション

第 14 回 一貫性(講義 90 分)

大規模な分散アプリケーションでは、共有データへの書き込みが性能上のボトルネックになりやすい。スケーラビリティを向上させるためには緩やかな一貫性という考え方が必要であることを理解し、その適用事例を学ぶ。そして、データの一貫性と深い関係にある分散システムの時間モデルを学び、メモリの一貫性を表現するモデルを学ぶ。

(1) 緩やかな一貫性

1. 厳格な一貫性の限界
2. TTL を用いる方式(DNS、Web Proxy など)
3. ゴシッププロトコル
4. ユーザに委ねる方式(CVS など)

(2) 分散システムの時間モデル

1. 時間の整合性
2. Happens before 関係
3. 論理時計

(3) 一貫性モデル

1. 契約としてのメモリー貫性モデル
2. Sequential consistency
3. Release consistency

第 15 回 大規模な分散システムの事例(講義+ワークショップ 120 分)

今日の分散情報システムには、小さな LAN 内に限定されるものから全地球規模のものまでがあり、システムのスケールによって設計方針やアーキテクチャの妥当性に対する考え方も異なってくる。代表的な大規模システムの事例について学び、設計方針の妥当性に関連した議論を行う。

(1) 事例研究

1. Akamai のコンテンツ配信ネットワーク、Google File System と Bigtable など

(2) 設計に関する議論(ワークショップ)

1. 要求分析(スケーラビリティへの要求、耐故障性への要求、利便性への要求、コストへの要求、セキュリティへの要求など)
2. トレードオフの考察

以上