2010/2/11版

クラウドコンピューティングのインパクト

2010年2月8日
於いて　公立はこだて未来大学

宮西洋太郎
(元　公立はこだて未来大学教授)

http://www.jade.dti.ne.jp/~miyanisi/index.files/impactofCC.pdf

この背景画は、「レンガのように、知識を積み上げよう」を意味する

若いみなさんに申し上げたいこと(1)

1. **大学では、大学でしかできないことに専念する**
   それが、生涯のスパンで見ると、最も効率的な時間の使い方
   (具体的には、学生時代のアルバイトは最小に、勉強は最大に)
2. **情報技術は進化してやまない**
   他の分野よりも進歩が激しい(コンピュータの歴史は新しい)
   ・ジェームズワットの蒸気機関の発明は1769年、今から約240年前
   ・カールベンツの自動車エンジンの発明は1879年、今から約130年前
   ・ホイートストーンの有線電気通信は1839年、今から約170年前
   ・マルコーニの無線電気通信は1901年、今から約110年前
   それに比べ、現代のコンピュータの発明は、約60年前
3. **自己の価値を維持**
   環境の進化に追従し、自己も進化する、さもないと、自分の価値が下がる
4. **青春の詩** 「求めて止まぬ探求心、人は信念と共に若く・・・、サミュエル・ウルマン」
   言語にしても、ツールにしても、方法論にしても、新しい技術がでてきたとき、「めんどうくさい」と感じたら、技術者として老人の域に入ったものと思え、新技術を追従していくのは、情報技術者が現役情報技術者たるための宿命である(自戒をこめて)
5. **文理両道(ブンリリョウドウ) 理実兼行(リジツケンコウ)**
   文系の知識、理系の知識の両方に通じ、理論と実践の両方をこなす(理論の知識は長持ちする、実践のスキルは長持ちしないが即戦力になる)
   理系の学生にこそ気づいてほしい。世の中の大枠は文系、各論で理系。
   昔の武士は、文武両道(頭脳と体力の鍛錬)

若いみなさんに申し上げたいこと(2)

青春の詩　「求めて止まぬ探求心、人は信念と共に若く・・・、サミュエル・ウルマン」

青春とは人生のある期間を言うのではなく心の様相を言うのだ。
優れた創造力、逞しき意志、炎ゆる情熱、怯懦を却ける勇猛心、安易を振り捨てる冒険心，こう言う様相を青春と言うのだ。
年を重ねただけで人は老いない。理想を失う時に初めて老いがくる。
歳月は皮膚のしわを増すが情熱を失う時に精神はしぼむ。
苦悶や、狐疑、不安、恐怖、失望、こう言うものこそ恰も長年月の如く人を老いさせ、精気ある魂をも芥に帰せしめてしまう。
年は七十であろうと十六であろうと、その胸中に抱き得るものは何か。
曰く「驚異への愛慕心」、空にひらめく星晨、その輝きにも似たる事物や思想に対する欽迎、事に處する剛毅な挑戦、小児の如く求めて止まぬ探求心、人生への歓喜と興味。
人は信念と共に若く、人は自信と共に若く、希望ある限り若く、疑惑と共に老ゆる、恐怖と共に老ゆる、失望と共に老い朽ちる。
大地より、神より、人より、美と喜悦、勇気と壮大、偉力と霊感を受ける限り人の若さは失われない。
これらの霊感が絶え、悲歎の白雪が人の心の奥までも蔽いつくし、皮肉の厚氷がこれを固くとざすに至ればこの時にこそ人は全くに老いて、神の憐れみを乞う他はなくなる。

情報技術の進歩（ハードウェア）

- リレー（継電器）によるディジタルコンピュータ
- 真空管によるディジタルコンピュータ
- トランジスタによる（ディジタル）コンピュータ
- ICによるコンピュータ
- LSIによるコンピュータ
- ダウンサイジング（どんどん小さくなる）の流れ
  メインフレーム（大型コンピュータ）から
  ミニコン、オフコン、パソコンへ
- マイクロプロセッサ、パーソナルコンピュータ（PC、パソコン）
- 個人が所有するコンピュータ
- 機器に組み込まれたコンピュータ（携帯電話、ゲーム機、家庭電気製品、自動車など）
- 別の流れは、並列コンピュータ、スーパーコンピュータ、グリッドコンピュータなど

情報技術の進歩（ネットワーク）

- 専用線によるコンピュータネットワーク
- 電話線によるコンピュータネットワーク
- インターネットの出現（ARPAネット）
- LANの出現、高速化　物理層IEEE802.3
  10Mbps→100Mbps→1000Mpbs
- プロトコルの自然淘汰（TCP/IPへ）
  ISO対TCP/IP→TCP/IPに淘汰された
- 広域ネットワークの帯域幅の拡大（ブロードバンド）
  アナログ回線→ADSL→光ファイバー
- 無線通信ネットワーク、高速化　物理層IEEE802.11

情報技術の進歩 (ソフトウェア)

- マシン言語によるプログラミング
- アセンブラ言語によるプログラミング
- コンパイラ言語、インタープリタ言語によるプログラミング
- ソフトウェア危機 (Software Crisis)
- ソフトウェアエンジニアリング (Software Engineering) の誕生
- 構造化プログラミング
- 大規模ソフトウェアの開発技法の整理 (ウオータフォール開発プロセス、構造化分析設計技法)
- オブジェクト指向、データ中心
- サービス指向

情報システム構築技法の変遷 (1)

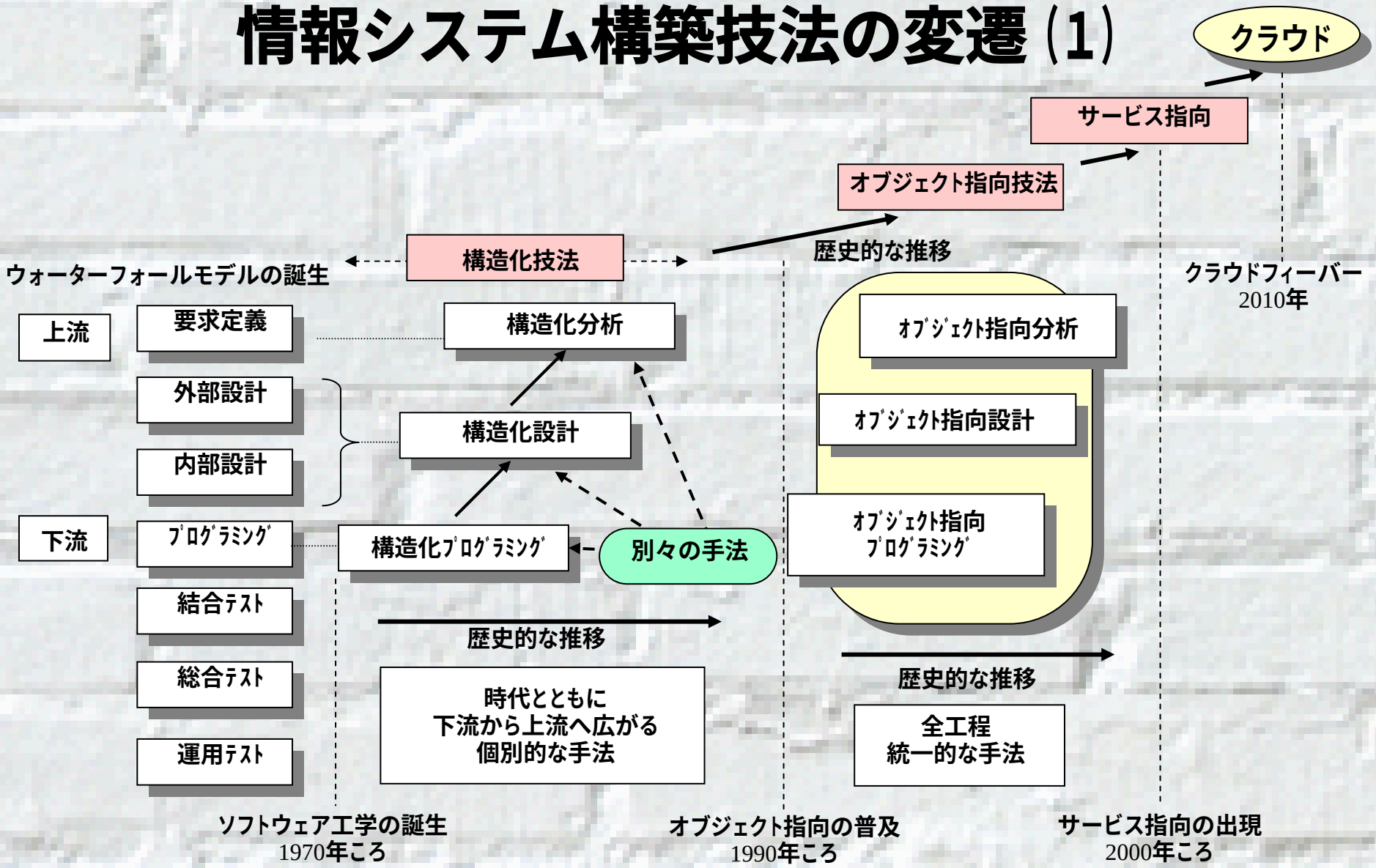


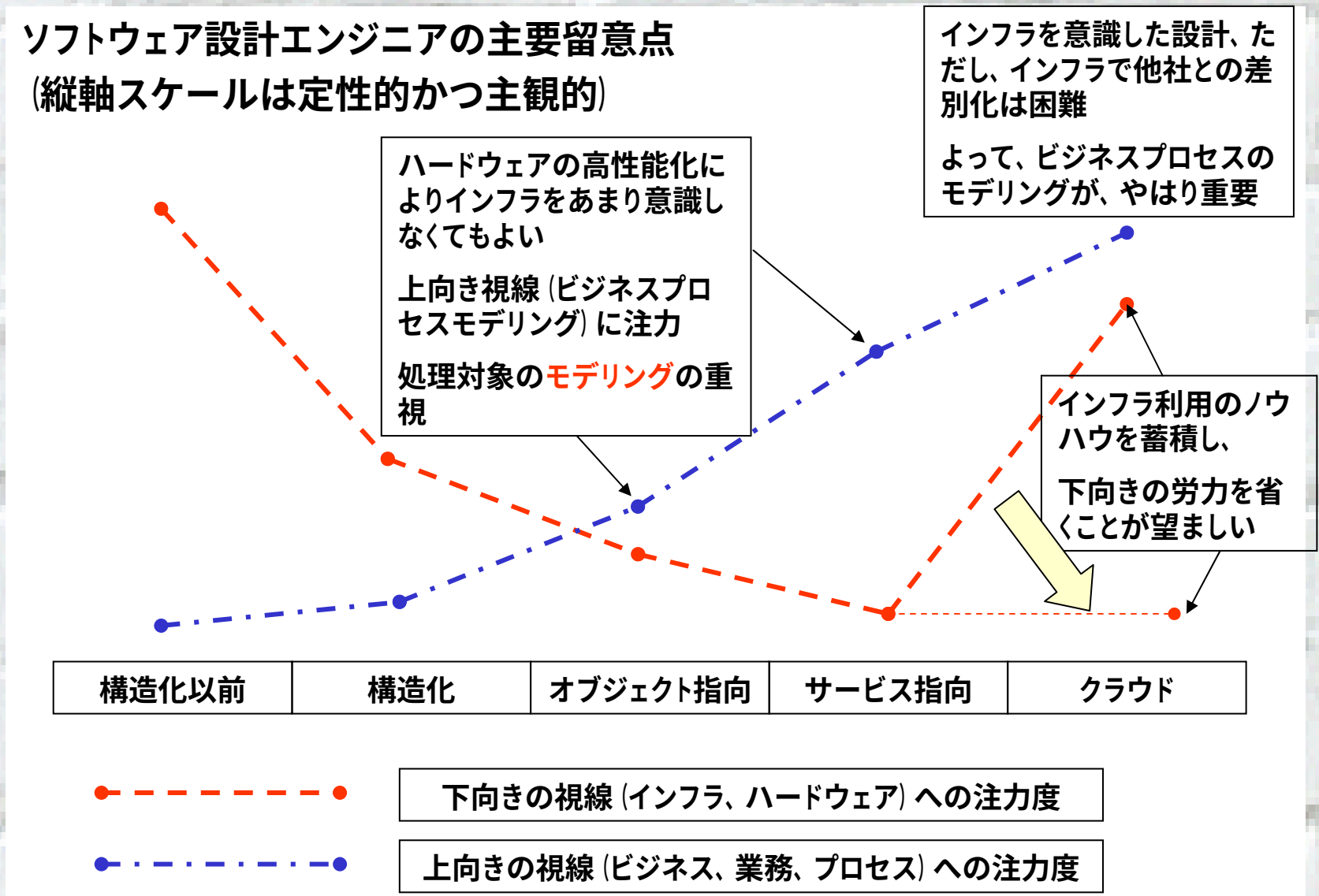
図 ソフトウェア工学の誕生から、構造化技法、オブジェクト指向技法、サービス指向、クラウドへ

情報システム構築技法の変遷 (2)

ソフトウェア設計エンジニアの主要留意点

1. ソフトウェア工学以前 (Before Software Engineering)の時代
   ハードウェアリソース (主メモリの容量、実行ステップ数) に留意。
2. 構造化の時代
   スパゲッティプログラムの排除 (goto less)、ソフトウェアの構造に留意、「連接、選択、反復」のパターンにはめる。
3. オブジェクト指向の時代
   ソフトウェア体系の中を動き回る情報の塊 (オブジェクト) に留意、できるだけ自律的単位 (属性、メソッドをもつ) としてまとめ、不要な情報は隠蔽する。
   人間社会のモデリング (モデル化) 、情報の擬人化、そのモデリング。
4. サービス指向の時代
   オブジェクトの粒度をさらに大きく、サービスという単位で捉える。
   3～4の流れで、アプリケーションプログラムの視線は、上へ上へと、用途 (業務) のほうを向いてきた。ビジネスプロセスモデリングなど。
5. クラウドコンピューティングの時代
   再び、インフラを意識する必要がでてきた。

情報システム構築技法の変遷 (3)



情報システム構築技法の変遷 (4)

分散システム構築方法が目指してきた方向 (原則) の1つ

透過性 (Transparency)

分散していることを意識しなくても良い、位置透過性

オブジェクト指向が目指してきた方向 (原則) の1つ

情報隠蔽 (Information hiding)

オブジェクト内部のことは外にださない、自律動作

どちらも、上位レベルの機能を実現するために働いている下位レベルの事柄は、上位レベルから見え(意識し) なくてもよい。見えないようにする。

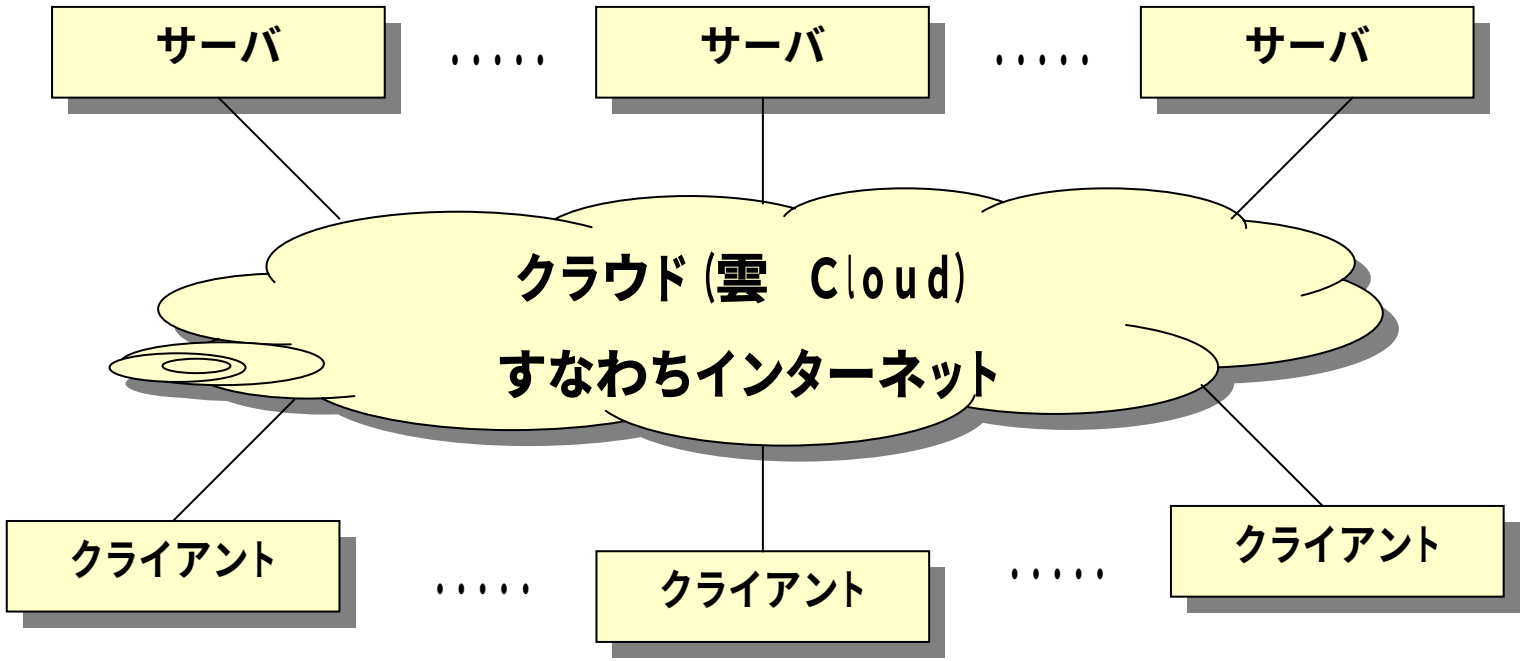
クラウドコンピューティングでは、

サーバの所在場所はユーザからまったく見えない。その意味では透過的である。ただしソフトウェア構築の際、インフラを意識する必要がある。

情報技術の進歩
(ソフトウェアアーキテクチャ)

- ホスト集中処理 (ホスト－周辺端末)
- ホスト－遠隔端末 (リモートターミナル)
- 分散処理、コンピュータネットワーク
- クライアント・サーバ
- 3層 (three tiered) クライアント・サーバ
  (クライアント＋サーバ＋バックエンド (DB))
- Webアプリケーション
  (クライアントとサーバとの組み合わせが地理的に自由になったが、論理的には固定的) (URLで接続)
- クラウドコンピューティング
  (クライアントとサーバとの組み合わせが完全に自由、クラウド側でサーバとの対応の面倒をみている)

クラウドコンピューティングとは



クライアントからの情報処理要求は、雲の上（インターネット上）にある、場所を意識しなくても良いどこかのサーバで処理されて、結果が戻ってくる。このような形態での情報処理をクラウドコンピューティングという

クラウドコンピューティングの メリット／デメリット

メリット

- 自社にサーバを設けなくてもよい（初期費用、維持費用の節減）
- スケール（ユーザ数、データ量などの規模）について気にしなくてもよい（従来とは桁違いのスケーラビリティをもつ）
- 広い範囲からサービス（機能）を借用することにより、開発期間を短縮できる可能性がある（開発の高速化）
- 総合的に安価に情報システムを構築できる可能性がある（使用期間にわたる総合費用の節減）
- 大規模計算を並列処理で高速化できる可能性がある（処理の高速化）

デメリット

- 外部にデータが漏れる恐れがある
- サービス利用に費用がかかる（運用費用）
- ブラックボックスの利用で、トラブル時の対応に不安
- ブラックボックスの利用で、テストのやりかたに注意が必要
- 従来のデータベース、トランザクションの考えを使えない
- インフラを意識したシステム構築が必要→ノウハウ獲得の要あり
- 任意の情報処理と言うより、元はと言えば、やはり大量の情報を検索するしくみから発生していることがいろいろな面での制約になっている（ようだ）

クラウドコンピューティングの要素技術

データベース

リレーショナルDB**から**KeyValue**データストア**（NoSQL）

トランザクション

ACID**トランザクションから**BASE**トランザクションへ**

並列処理

MapReduce

仮想化技術

1**つの物理的な**CPU**に複数の論理的な**CPU**を保持する**

複数の物理的なCPU**を**1**つの論理的な**CPU**として働く→分散処理**

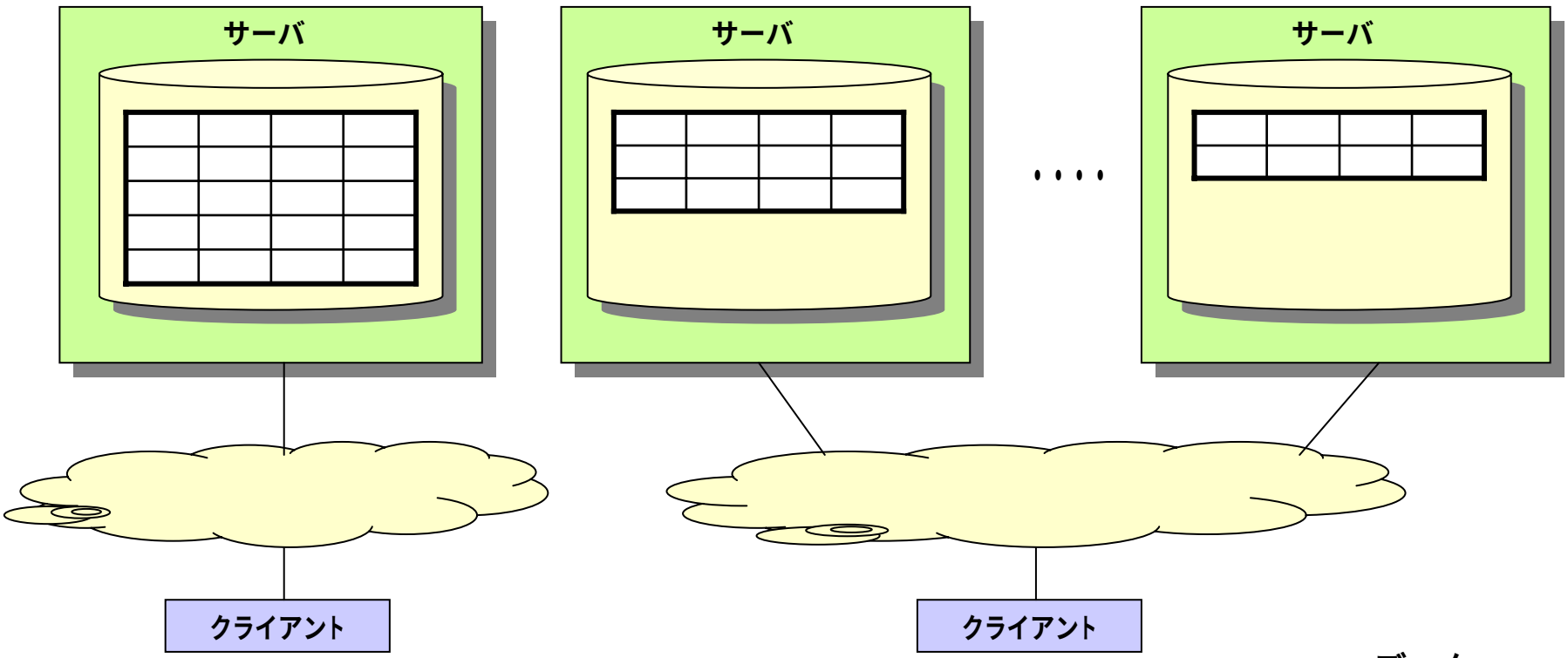
スケールアウト

担当サーバの割り当て、探索　（ハッシング）（コンシステントハッシング）

担当サーバへの到達、ルーティング（構造化オーバーレイ）

クラウドコンピューティングの要素技術

データベース　リレーショナルDBからKeyValueデータストア（NoSQL）



KeyValueデータ

i 行（タプル）	PK i	Vi1	Vi2	Vi3

基本的に、すべての行が1つのサーバに配置される（PK: Primary Key）

i 番目	Key i	Vi1	Vi2	Vi3

基本的に、この単位で、多数のサーバにバラバラに配置される

クラウドコンピューティングの要素技術

トランザクション

ACID (Atomicity, Consistency, Isolation, Durability)トランザクションから

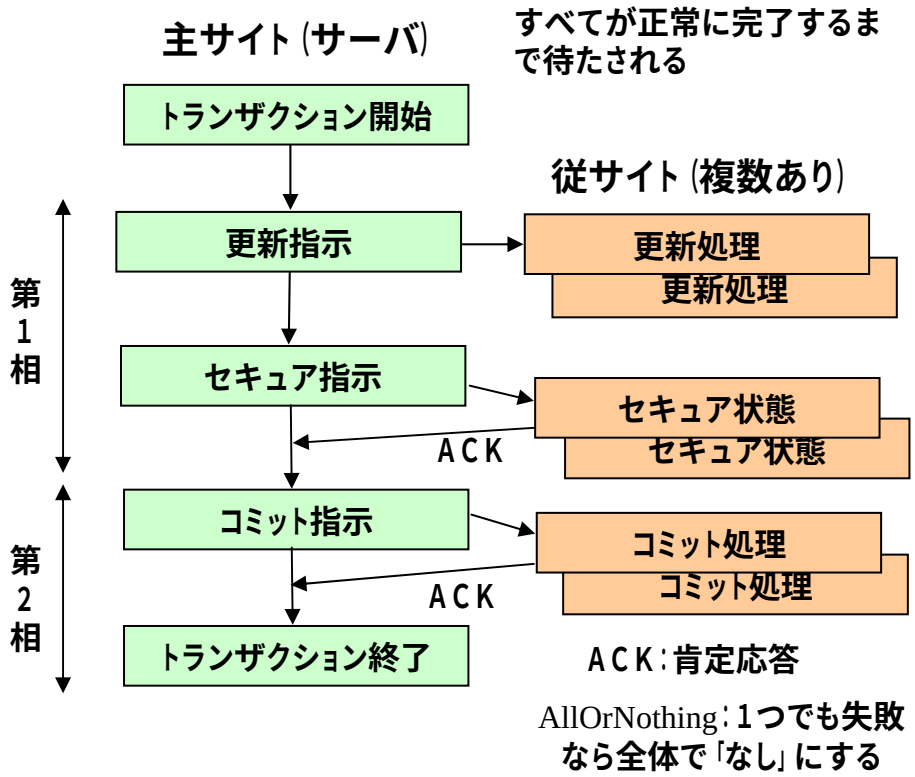
BASE (Basically Availability, Soft state, Eventually consistency)トランザクションへ

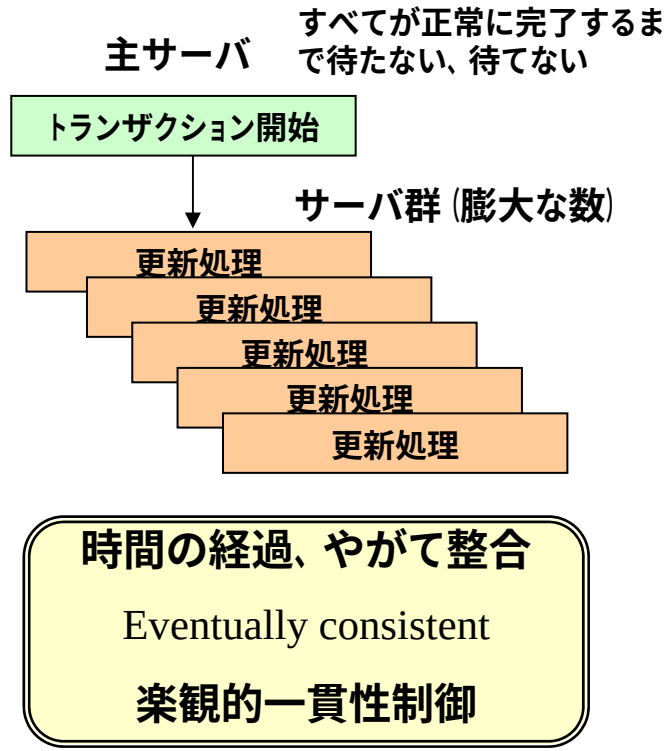
一貫性（整合性）の維持

強い一貫性制御方式 ACID, 2PC, WAL

一貫性（整合性）の維持

弱い一貫性制御方式 BASE



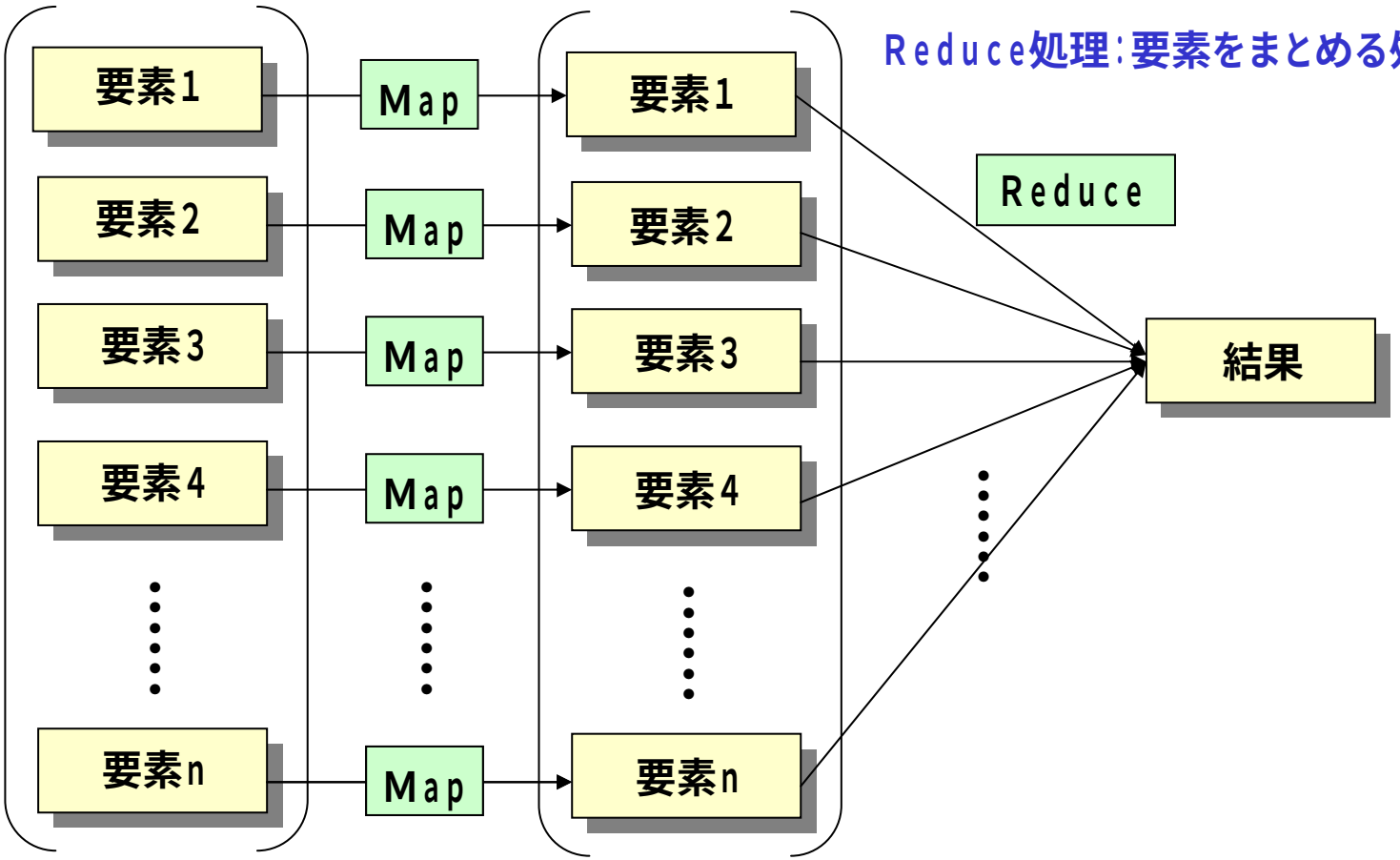


クラウドコンピューティングの要素技術

並列処理

MapReduce

Map処理：要素ごとに行う処理

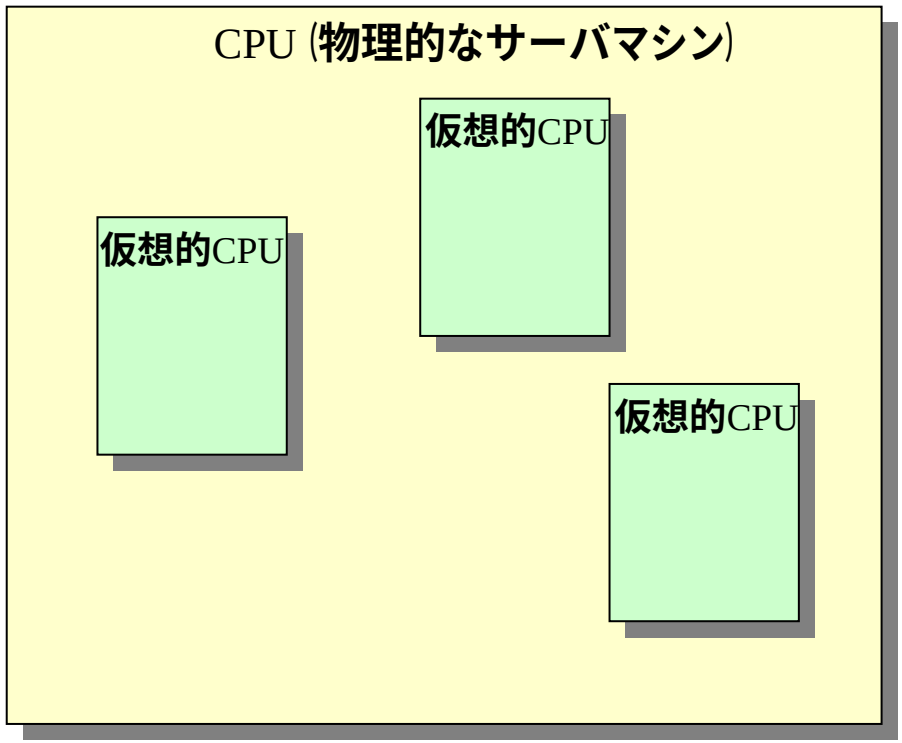


クラウドコンピューティングの要素技術

仮想化技術

1つの物理的なCPUに複数の論理的なCPUを保持する

複数の物理的なCPUを1つの論理的なCPUとして働く→分散処理

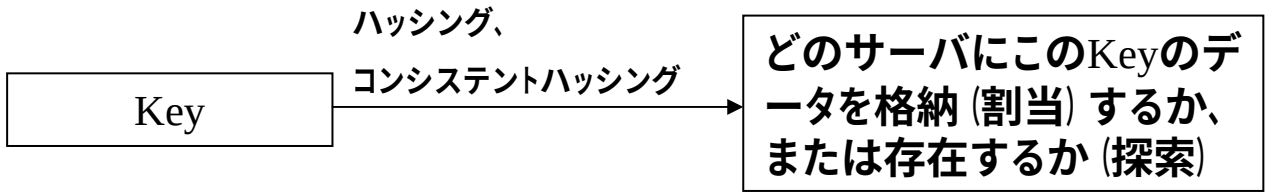


クラウドコンピューティングの要素技術

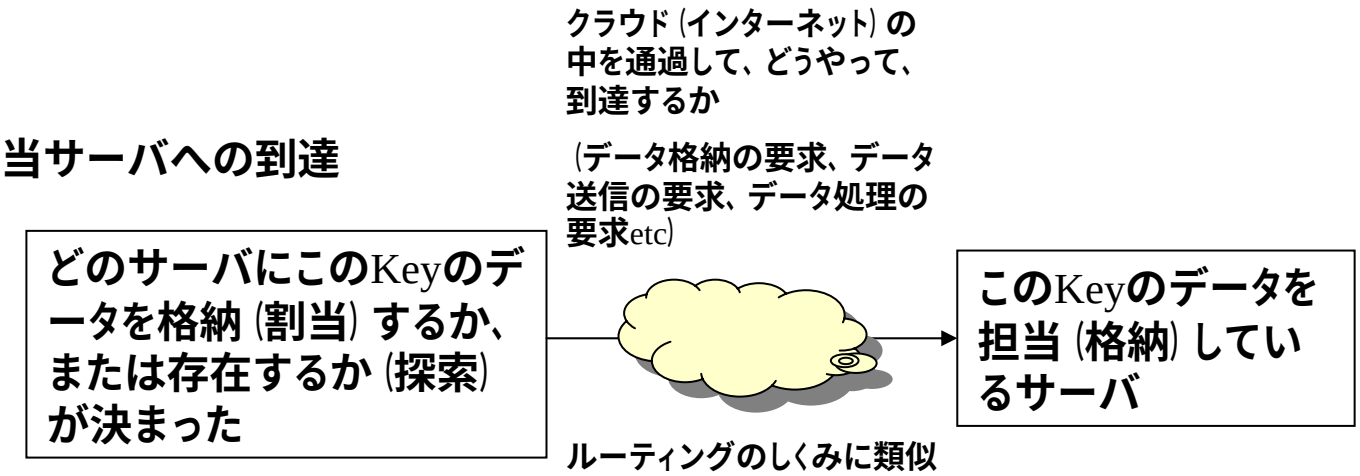
スケールアウト

担当サーバの割り当て、探索　(ハッシング) (コンシステントハッシング)
担当サーバへの到達、ルーティング (構造化オーバーレイ)

担当サーバの割り当て、探索



担当サーバへの到達



クラウドコンピューティングサービス提供者

マイクロソフト （PaaS：Platform as a Service）

.Net、Azure

グーグル （PaaS）

Google App Engine

アマゾン （IaaS：Infrastructure as a Service）

Amazon EC2、 Amazon S3

セールスフォース （SaaS：Software as a Service）

Salesforce AppExchange

SaaS
PaaS
IaaS

クラウドコンピューティングによる情報システムの構築方法（1）

クラウドコンピューティングの要素技術は特徴でもあるが、情報システム構築の制約ともなる場合がある。

この制約をクリアしながら、業務用（エンタープライズ用）の情報システムを構築する手法、ノウハウを確立する必要がある。

特に

・データベース

・トランザクション

・並列処理の上手な利用

・セキュリティ

に注意が必要である。

・ひとつの解決方法は、制約にひっかからず、自社のコアコンピタンスではない部分のみ、クラウドにまかせ、それ以外は、自社内の情報システムに任せるというハイブリッドクラウドの考えもある。

・セキュリティについては、外部にデータを保存する危険性を避けるため、自社内にクラウドを構築するプライベートクラウドという考えもある。この場合も、会社規模が大きいほど、ハードウェアやソフトウェアの共通利用を図ることができ、メリットがある。これに対して、通常のクラウドをパブリッククラウドという。

参考文献(2)-7、(3)

基本的には、今後の課題

クラウドコンピューティングによる情報システムの構築方法（2）

日経SYSTEMS、2009年12月号記事から、参考文献(3)

- 要件定義フェーズでの留意点
  - 基盤（インフラ）を前提として、要件を定義する
    - 非機能要件（特に運用要件）（TIS中村健氏）
      - 運用要件をまとめ運用ポリシーとする
      - 運用ポリシーとサービス提供者のサービスレベルとの対比
      - フィット／ギャップ、ギャップの少ないサービス提供者を選択
      - ギャップは、運用要件をさげる、
    - 機能要件
      - 公開されているサービスの利用（TIS中村健氏）
      - アドオン開発
        - データモデルの設計（グルージェントの岡本聰氏）
          トランザクションの粒度（かたまりの大きさ）に要注意
          RDBからKeyValue型へ、データベースのアクセス時に集計は、あらかじめ集計しておく

クラウドコンピューティングによる情報システムの構築方法（3）

日経SYSTEMS、2009年12月号記事から、参考文献(3)

- 設計フェーズでの留意点
  - クラウドの特徴、制約に留意する
    - スケーラブルの特徴をいかす（インテック・ネットコア中川郁夫氏）
      1. KeyValueデータストアを並列にアクセス
      2. トランザクションの単位を小さくする
      3. 参照専用（書き込みの競合を防ぐ）
      4. 同一処理を複数実行、成功したものを採用
      5. セッション状態をもたない
    - セキュリティ問題の回避（テラスカイ竹澤聡志氏）
      - 処理は外部にさせて、結果を社内に保存
    - バックアップ方式（アビームコンサルティング関川秀一郎氏）
      - 複数のバックアップを併用

クラウドコンピューティングによる情報システムの構築方法（4）

日経SYSTEMS、2009年12月号記事から、参考文献(3)

・実装・テストフェーズでの留意点

実装フェーズ

– 制約（伊藤忠テクノソリューションズ高島元氏）

1. 実行ステートメント数の制限
2. クエリー発行数の制限

テストフェーズ

– テストの自動化（日立製作所宮崎肇之氏）

1. 回帰テスト（クラウド側のバージョンアップ時など）、テスト済の全アプリケーションを再度テストする
2. 繰り返しテスト、応答時間の測定など
3. 並列処理の確認
4. ログによる確認（グレーボックステスト）

チューニングの考え（クロスマーケティング竹下康平氏）

徐々にリソースを増やしていき、処理時間をみる

参考文献

(1)「日経コンピュータ」2009年9月2日号、特集：グーグル　次の一手
(2) 情報処理学会誌　2009年11月号（「情報処理」　Vol.50　No.11）
特集：クラウドコンピューティング
(2)-1 丸山不二夫「クラウドの成立過程とその技術特徴について」
(2)-2 中田秀基「Googleのクラウド技術」
(2)-3 石田愛「Amazon　EC2」
(2)-4 藤田昭人「クラウド技術とオープンソース」
(2)-5 首藤一幸「スケールアウトの技術」
(2)-6 山下哲也「クラウドとモバイルデバイス」
(2)-7 萩原正義「クラウドアプリケーションの分析と開発手法」
(2)-8 浦本直彦「クラウドコンピューティングにおけるセキュリティとコンプライアンス」
・上記のうち、エンタープライズ情報システム開発に最も直接的なものは、(2)-7。
・そのための予備知識として、(2)-1から(2)-3、(2)-5。
・クラウドコンピューティングを有償のベンダーサービスではなく、オープンソースで実現する方法が(2)-4。
・クラウドコンピューティングで懸念となるセキュリティを扱っているのが(2)-8。
(3)「日経SYSTEMS」　2009年12月号
特集：クラウド革命に備えよう
(4) ニューヨークだより(IPA)2009 年 9 月
「クラウドコンピューティングの産業構造とオープン化を巡る最近の動向」
http://www.ipa.go.jp/about/NYreport/200909.pdf
(5) クラウド活用による経営システム改革 ～NECの実践とソリューション～
http://www.nec.co.jp/effort/strategy/2009_1105/index.html
以下(6)～(13)、日経SYSTEMS(3)の紹介資料
(6) 米持幸寿、「クラウドを実現する技術」、インプレスジャパン
(7) 城田真琴、「クラウドの衝撃」、東洋経済新報社
(8) 日経BP社出版局編、「クラウド大全」、日経BP社
(9) Nicholas G. Carr、村上彩訳「クラウド化する世界」、翔泳社
(10) (株)グルージェント、「Google App Engine for Java [実践]クラウドシステム構築」技術評論社
(11) 並河祐貴、安達輝雄、「クラウドAmazon EC2/S3のすべて」、日経BP社
(12) 西田宗千佳、「クラウドの象徴セールスフォース」、インプレスジャパン
(13) マイクロソフトエバンジェリストチーム、「.NET開発テクノロジ入門」、日経BPソフトプレス
以下(14)は、情報処理(2)-2の紹介資料
(14) 西田圭介、「Googleを支える技術－巨大システムの内側の世界」、技術評論社(2008)

おわりに

本日お話したクラウドコンピューティングは、本当に、企業用または行政用の情報システムに役立つものかどうかを見極めましょう、

もし、そうならば、そのための知識と実践方法を身につけましょう。

一般論では、世の中の変化に常に敏感で、世の中の変化に取り残されないようにしましょう。

さらに願わくば、皆さんが世の中の変化を惹き起こす原動力になられんことを。

皆さんの、大学におけるますますのご精励と、社会にでてからのご活躍を、お祈りします。

ご清聴ありがとうございました。