

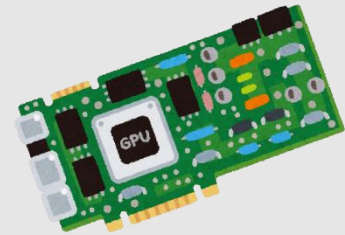
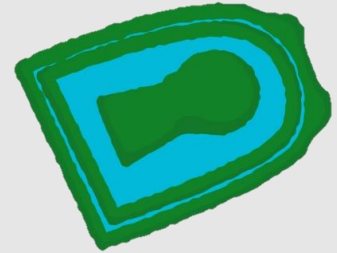
Cloudから始まる AIとBig Data

Kobe Spiral2020

佐伯幸郎

自己紹介

- 佐伯幸郎（さいきさちお）
 - システム情報学研究科 特命講師
 - カーブと古墳とビデオカード
 - お酒は嗜む程度
- 文部科学省の実践的教育プログラムを担当
 - IT Spiral（2006～2012）
 - ソフトウェア工学
 - Cloud Spiral（2012～2017）
 - ソフトウェア工学＋クラウド技術
 - AiBiC Spiral（2016～現在）
 - クラウド技術＋AI/BigData



Cloud  Spiral

AiBiC  Spiral

普段の活動

- 様々な企業と共同でPBLを通じた実践的な教育を大学院生・学部生に行う

amazon

IBM

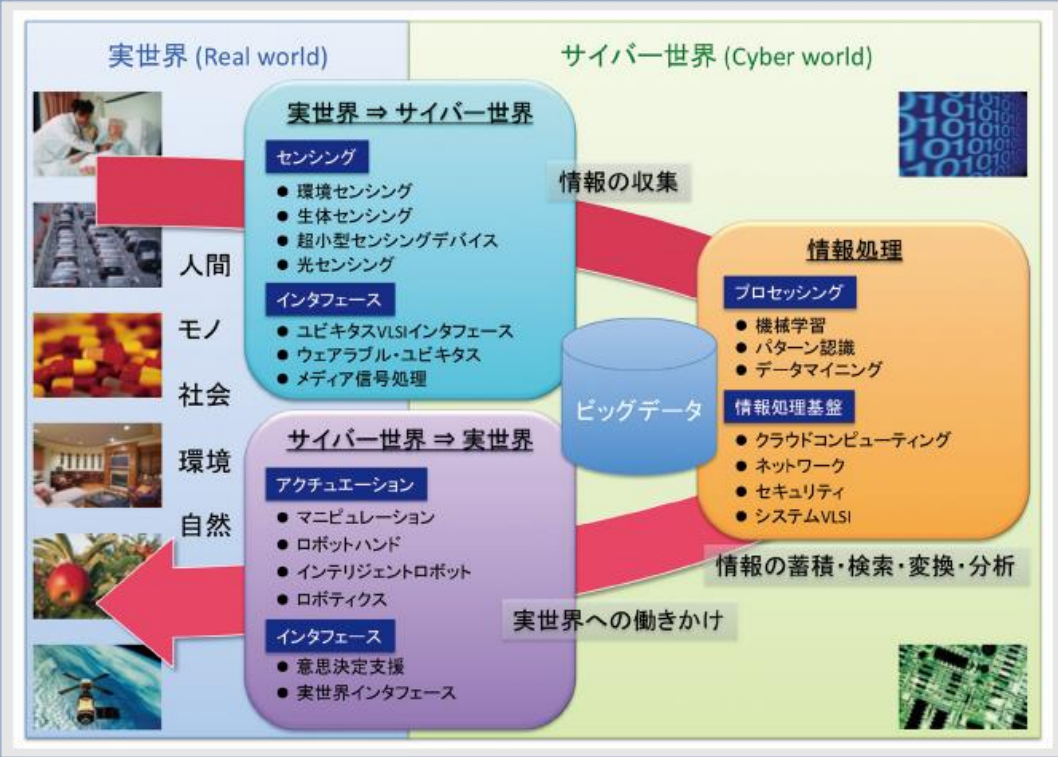
Microsoft

Rakuten



今日の目的 (表)

超スマート社会・Society5.0など言われる昨今のサイバー世界を支えるビッグデータ・AI技術が近年なぜ急速に発展したのかをクラウドを視点として知る



今日の目的(裏)

案外みんなクラウドを知らない

最初に

今日の講義は学術・研究的な側面は無い

- 自身の知識を広げる場だと思って聞いて下さい
- $A:B:C = 0.5:1:8.5$

技術の発展は複合的な要因が多い

- あくまで一つの解釈例だと思って下さい

質問はいつでも受け付けます

- 分からないところがあれば途中で止めて下さい

最後にちょっとした課題をします

AI

なぜ並ぶのか？

Big Data

Cloud

目次

- AI 概論？
- BigData 概論？
- Cloud 概論

AI概論？

AI (Artificial Intelligence) とは

定義：

- 知能・知識とは
- 計算機における知能

歴史：

- 第1次人工知能ブーム
 - 冬の時代
- 第2次人工知能ブーム
 - 冬の時代
- 第3次人工知能ブーム

AI (Artificial Intelligence) とは

定義：

- 知能・知識とは
- 計算機に

歴史：

- 第1次人工知能ブーム
 - 冬の時代
- 第2次人工知能ブーム
 - 冬の時代
- 第3次人工知能ブーム

田略

「Deep Learning」の検索トレンド

最近はやりのAI = Deep Learning

人気度の動向



なぜAI (DL) が急激に広がったのか

- 計算機の性能向上
 - メモリ・CPU・GPU



- データの充実
 - Deep Learningはデータによる**数の暴力**（暴論）
 - NNの勾配消失問題への対応



なぜAI (DL) が急激に広がったのか

- 計算機の性能向上
 - メモリ・CPU・GPU



- データの充実
 - Deep Learningはデータによる**数の暴力**（暴論）
 - NNの勾配消失問題への対応



Big data概論？

Big Data?



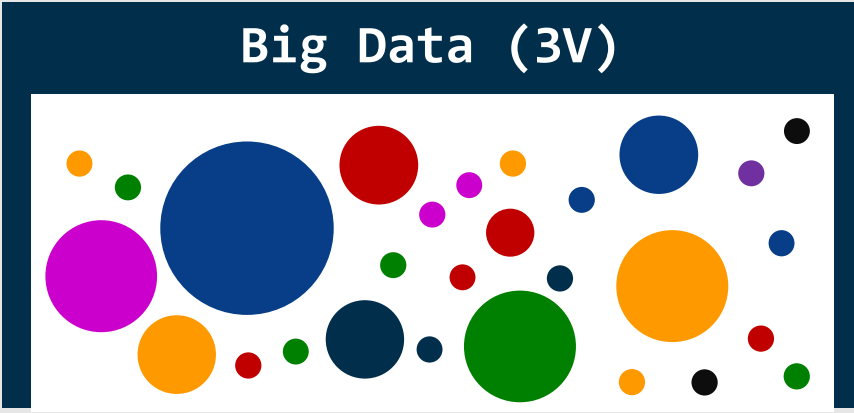
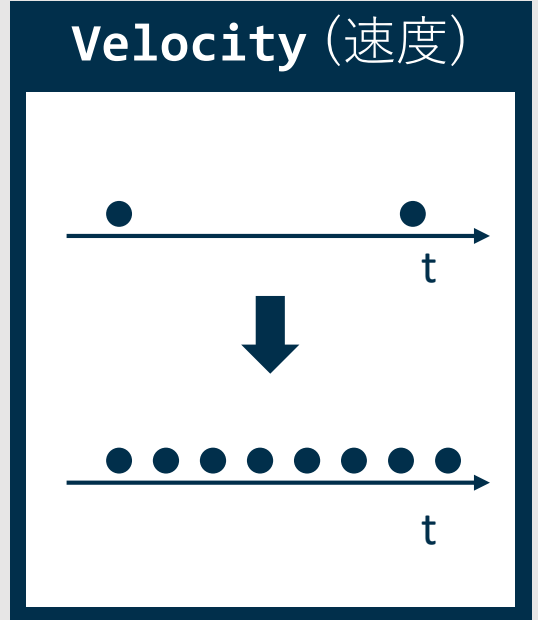
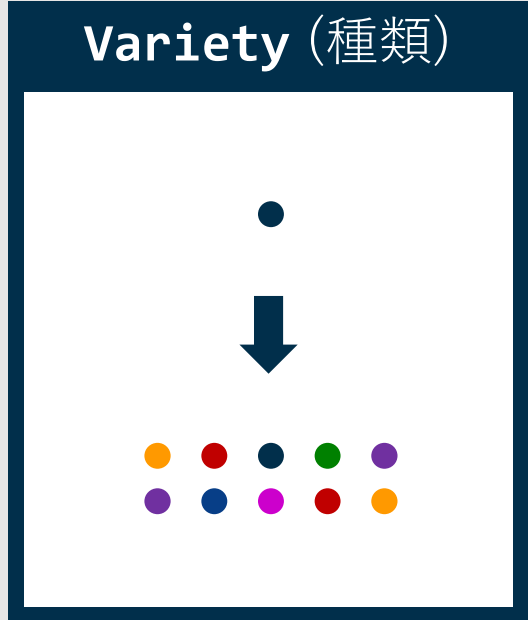
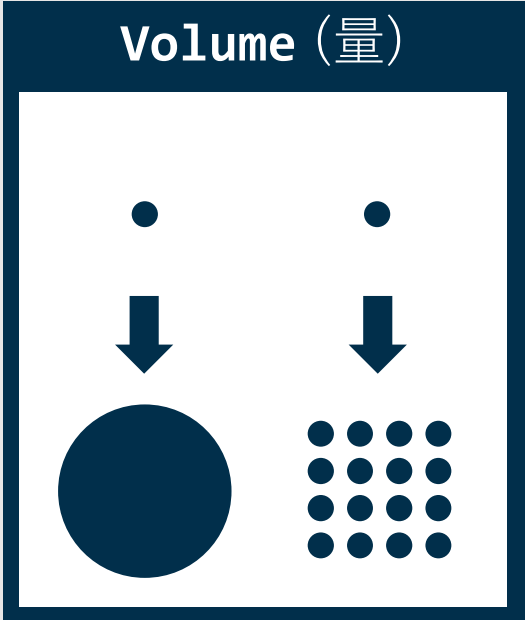
「BigData」の検索トレンド

人気度の動向



注

ビッグデータ = 3Vを満たすデータのこと



ビッグデータの活用事例 1/2

- Amazon

- 大量の購入履歴・閲覧履歴を使った宣伝
(推薦システム)

この商品を見た後に買っているのは？



ASUS Nexus7 (2013) TABL

★★★★☆ (95)

¥ 27,000

よく一緒に購入されている商品



+



+



対象商品: ASUS Nexus7 (2013) TABL
ME571-16G ¥ 24,800

- 国土交通省

- 橋に大量のセンサーを設置
- ひずみ, 振動, 劣化状況などを計測
- 交通インフラの老朽化を迅速に察知

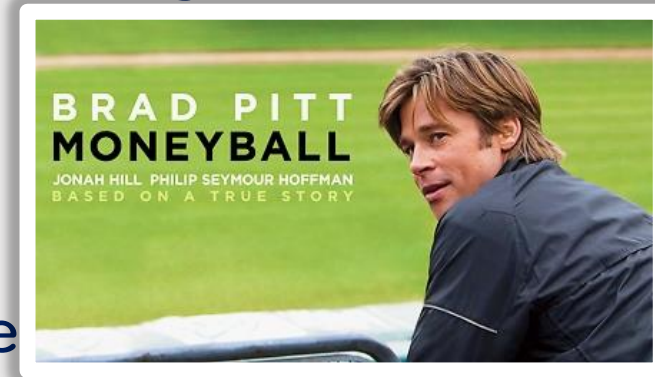
ビッグデータの活用事例 2/2

貧乏&弱小球団 アスレチックス

- データに基づく **チーム作成**
 - 選手の能力をデータ化, **コスパの良い選手**をスカウト
- 経験や勘によるバイアスを完全に排除
 - ~~才能がある~~, ~~今後伸びる~~, ~~フォームが悪い~~

オバマ大統領

- データに基づく **選挙キャンペーン**
 - 「オバマだけど質問ある？」 @Re



I am Barack Obama, President of the United States -- AMA (self.IAmA)

PresidentObama が /r/IAmA に 1 年前 * 投稿

Hi, I'm Barack Obama, President of the United States. Ask me anything. I'll be taking your questions for half an hour starting at about 4:30 ET.

Proof it's me:

意味の変わりつつある「ビッグデータ」

- ~~3Vを満たすデータ~~ → データを活用した取り組み
 - データを計測して分析し、そこから価値を生む行為



- 意味のない議論
 - 何GBあれば/何種類あればビッグデータなのか？
- 基本はデータ計測 + データマイニング
 - 巨大になりがちなので工夫が必須

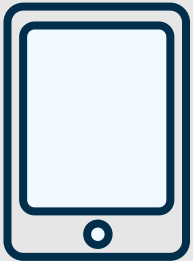
何でもデータ化できる時代



会員情報
アクセスログ
サービス利用状況



電子マネー履歴
POS履歴



メール
GPS
温度
湿度
加速度



ログイン時間
単位取得状況
アンケート
進路希望



つぶやき
写真
動画

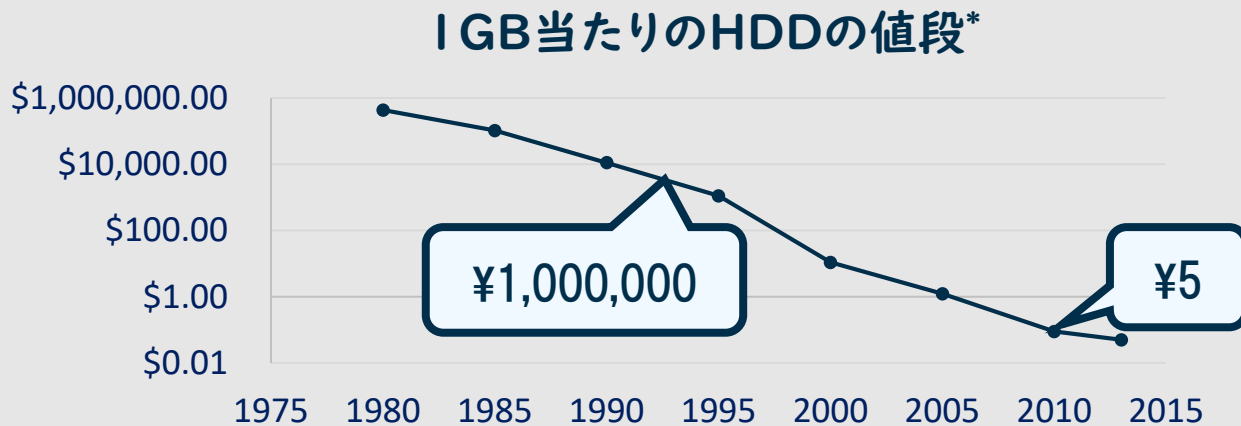


交通状況
道路状況
橋の状態



何でも蓄積できる時代

- ストレージの低価格化・大容量化



- NoSQLの登場
 - RDBは高機能なクエリ言語 (SQL) を持つが大量データに弱い
 - DBとしての厳格さやSQLを捨てる代わりに高速化・大規模化

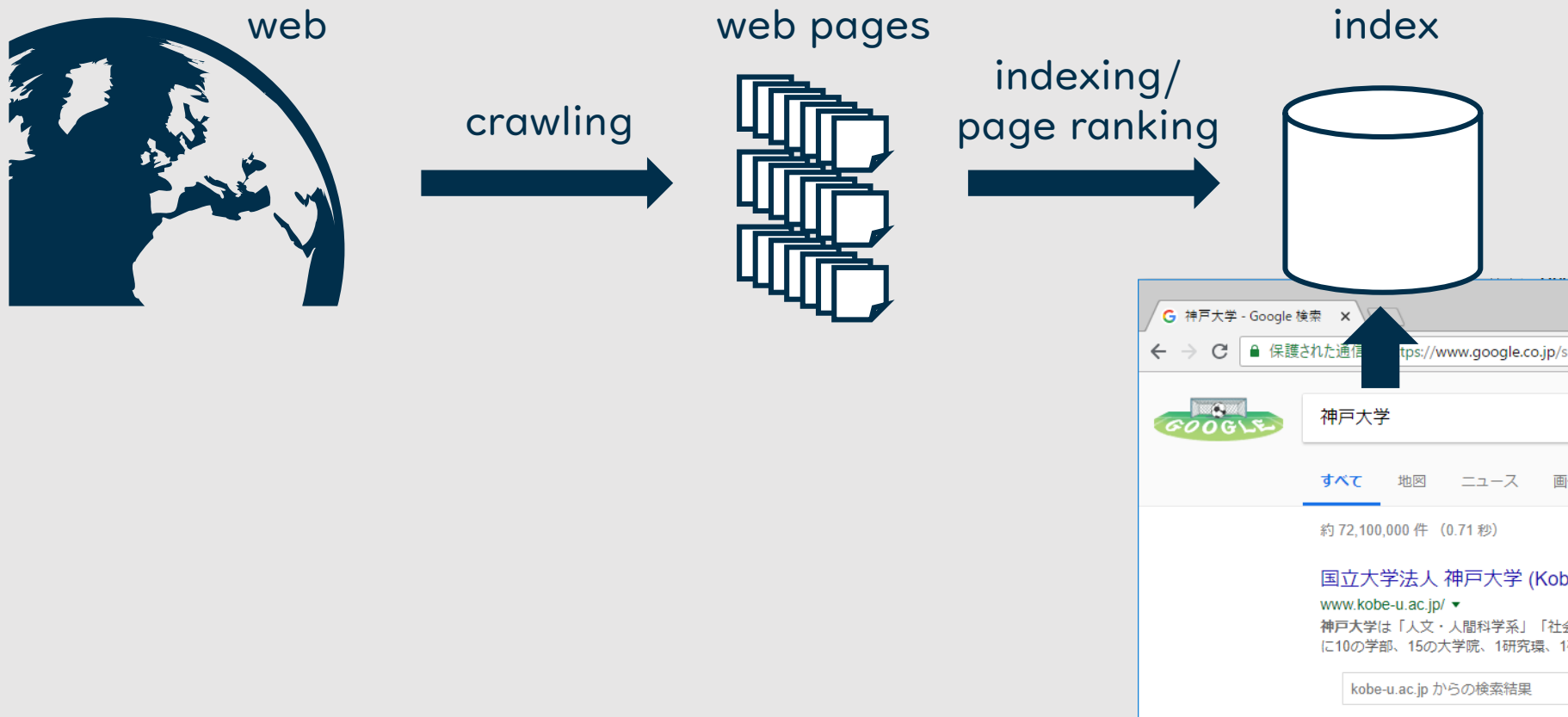
どう処理するか？

- 蓄えたデータをどう処理するか？



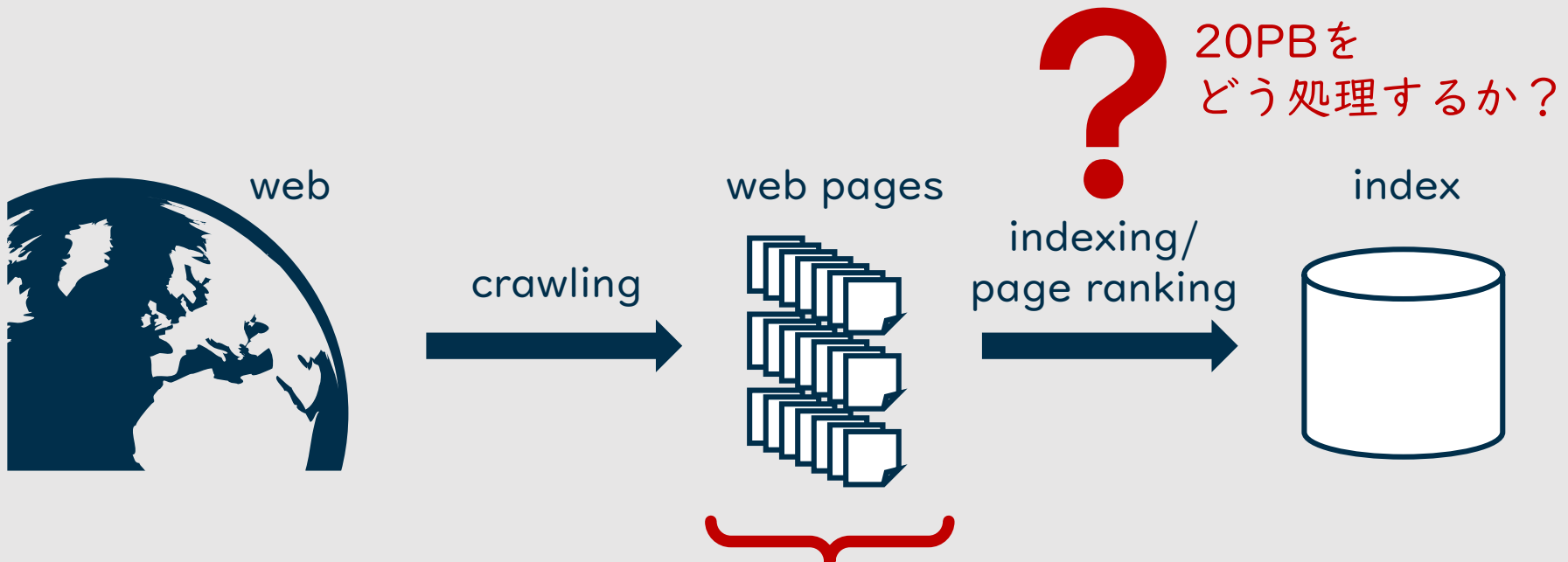
Googleの場合

- 検索エンジンは正にビッグデータ処理



Googleの場合

- 検索エンジンは正にビッグデータ処理



20,000,000,000,000,000 bytes / day !*

20PB. 1台のHDD (SATA3接続) だと読み込みだけで400日弱

* J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, Comm. ACM, vol.51, no.1, pp107-113, 2008

分散しよう

- 100台で処理すれば100倍の速度！
 - 複数マシンで**分散処理**すればよい

スケールアウト

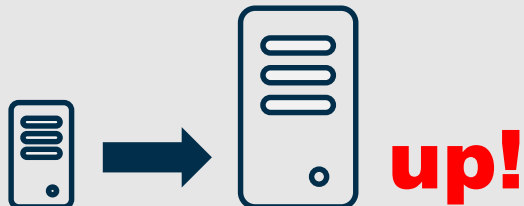
- 複数台並べて高速化



← 世の中の潮流はこっち

スケールアップ

- マシンの性能を上げて高速化



分散処理は難しい

- 分散環境の管理

- 障害をどう検知するか？
- ファイルをどう分散配置するか？
- タスクをどうスケジューリングするか？



- プログラムの開発

- データのどこからどこまでをどのワーカが担当するか？
- ワーカの同期をどうするか？
- 処理中にワーカが故障したときにどうするか？

- 開発者が**処理の本質以外**を気にかける必要あり

FortranとMPIによる率直な分散処理

- 10000までの**総和**を並列計算するプログラム

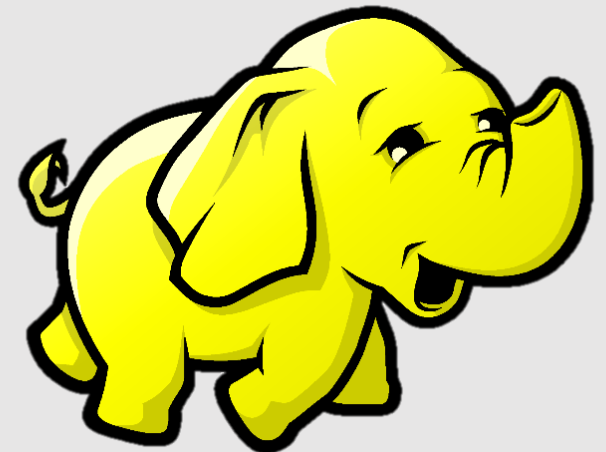
```
program sumn
  use mpi
  implicit none
  integer :: n,i,istart,iend,isum,isum1
  integer :: nprocs,myrank,ierr
  call mpi_init(ierr)
  call mpi_comm_size(MPI_COMM_WORLD,nprocs,ierr)
  call mpi_comm_rank(MPI_COMM_WORLD,myrank,ierr)
  if (myrank==0) n=10000
  call mpi_bcast(n,1,MPI_INTEGER,0,MPI_COMM_WORLD,ierr)
  istart=n*myrank/nprocs+1
  iend=n*(myrank+1)/nprocs
  isum=0
  do i=istart, iend
    isum=isum+i
  end do
  call mpi_reduce(isum,isum1,1,MPI_INTEGER,MPI_SUM,0,MPI_COMM_WORLD,ierr)
  if (myrank==0) print *, 'sum =', isum1
  call mpi_finalize(ierr)
end program sumn
```

本質

データのどこからどこまでを
だれが担当するかを決める

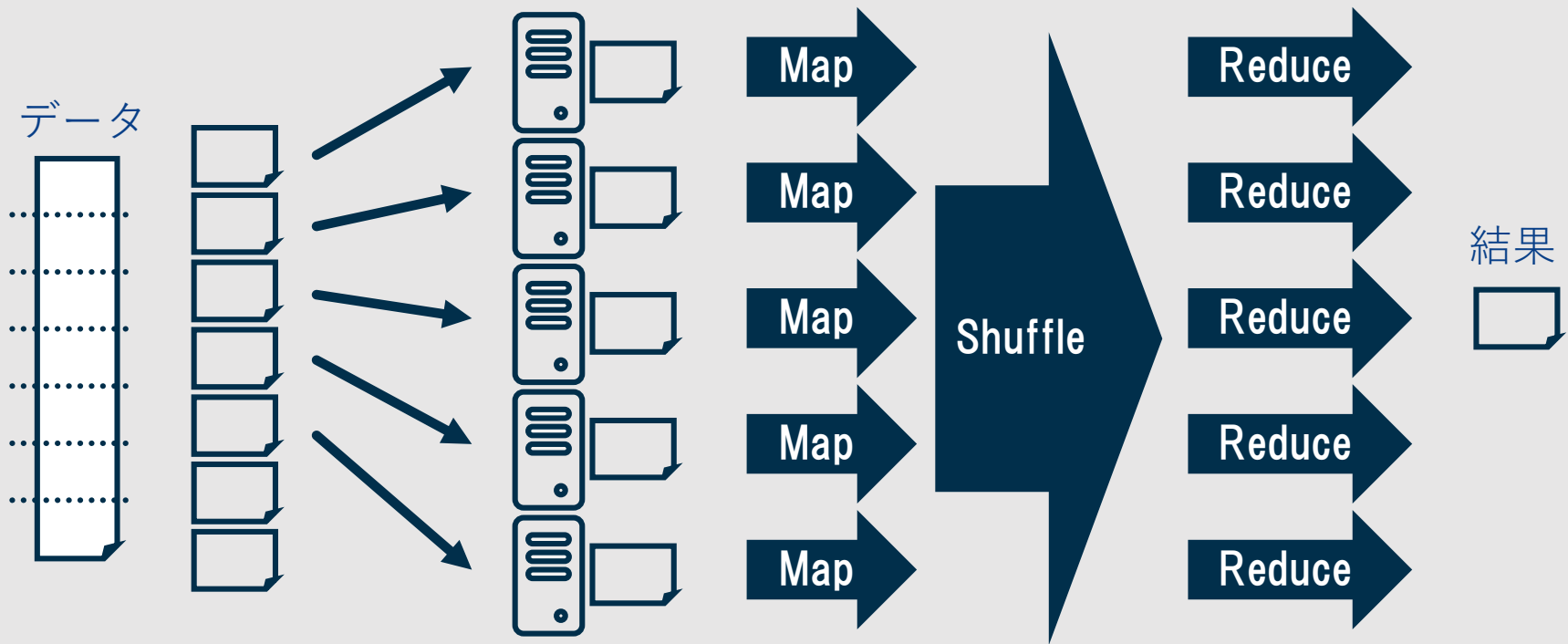
Hadoop MapReduce

- 大規模データの分散処理フレームワーク
 - = MapReduceを実行するためのフレームワーク
 - 分散処理の難しさを軽減してくれる
 - 開発者を処理の本質に集中させる
 - 手軽でオープンな分散処理環境
- 高い実績
 - Yahoo!, Facebook, IBM, Last.fm, . . .
- 2つの主要コンポーネント
 - HDFS & MapReduce



Hadoop全体像

- HDFSでデータを分散配置
- MapReduceでデータを分散処理

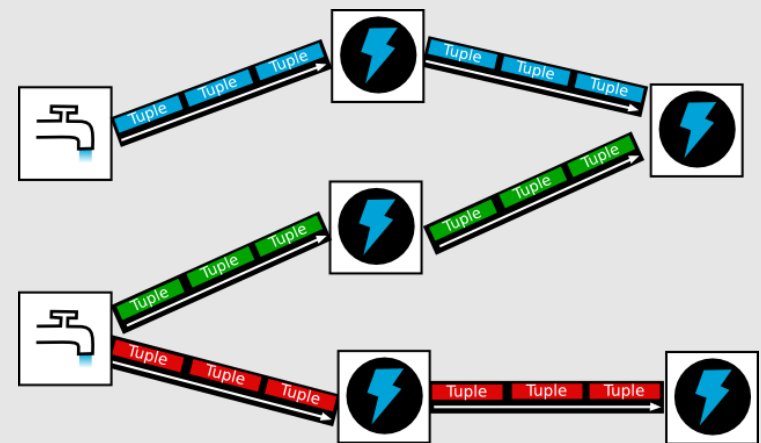


手段としてのHadoop

- Hadoopは**巨大データ処理方法の一つ**
 - Hadoopのシェアは大きいが...
 - 巨大なデータを処理できればなんでもよい

- 別の技術も生まれつつある
 - MRを生成するSQLライクな言語 (Pig)
 - リアルタイムMR (Presto)
 - Hadoop/MRとは全く別の分散処理技術 (Spark, Storm)

- **共通する概念はスケールアウト**
 - スケールアウトしない技術は変化する未来に追従できない



Sparkのアーキテクチャ

「ビッグデータ」はバズワード？

- チャラい言葉
 - 論文や科研の申請書等に乱用されてる
 - 言葉はなくなるかもしれないが**概念は残る**
 - むしろ当たり前になる
 - データに基づく意志決定・ビジネス戦略



- **データを集めやすい世の中の流れ**も後押し
 - スマホ，クラウド，Internet of Things

統計学とビッグデータ

- 統計学

- 全体を推測する方法
- 国勢調査や疫病調査が目的で生まれた手段
- あくまで推測なので「確率」と隣り合わせ

- ビッグデータ

- 全体・少数の両方を正確に把握できる方法
- 全てを計測して全てを発掘する
- (予測という問題以外は) 確率の入る余地がない
- 少数を正確に把握できるので個人最適化戦略が可能に



≡ ロングテール

Cloud 概論

なぜAIが急激に広がったのか

- 計算機の性能向上
 - メモリ・CPU・GPU



- データの充実
 - Deep Learningはデータによる数の暴力（暴論）
 - NNの勾配消失問題への対応



クラウド

「クラウド」の検索トレンド

人気度の動向

なぜこのような急成長と普及を果たしたか？



注

クラウドの語源

- 2006 Googleサーチエンジン戦略会議
 - 元CEO エリック・シュミット

ブラウザの種類もアクセス手段もパソコンかマックか携帯電話かも無関係です。**雲**のような、**巨大な**インターネットにアクセスすれば、その利益、その**恵みの雨**を受けられる時代になっています。

元々は大きさと恵みの雨というメタフェ



代表的クラウドサービス



facebook®



代表的クラウドサービス



- オンラインのデータ保存サービス
 - サービスの開始登録は数分で完了
 - 登録すれば無料で2GBまでのデータ保存可能
 - 課金契約すればもっとたくさん保存可能
 - いつでもすぐに解約可能

どうすれば低価格でこのようなサービスが実現できるのか？

キー要素は「仮想化ソフトウェア技術」

仮想計算機技術（仮想マシン）の確立

ソフトウェアで仮想的に構築された計算機環境

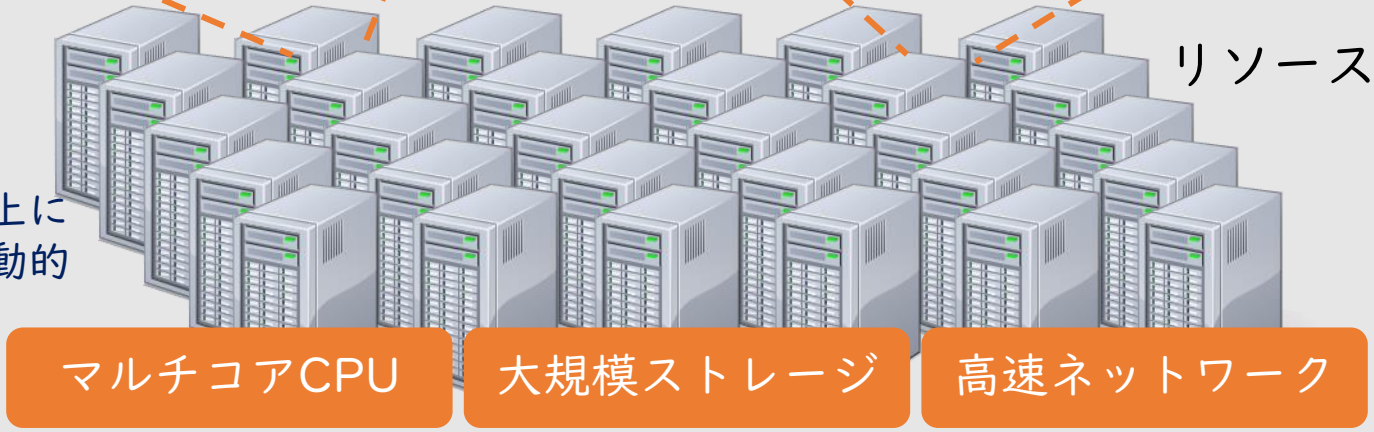


ソフトウェアで仮想的に構築された計算機環境



リソースプール

物理的な計算機上にソフトウェアで動的に環境構築



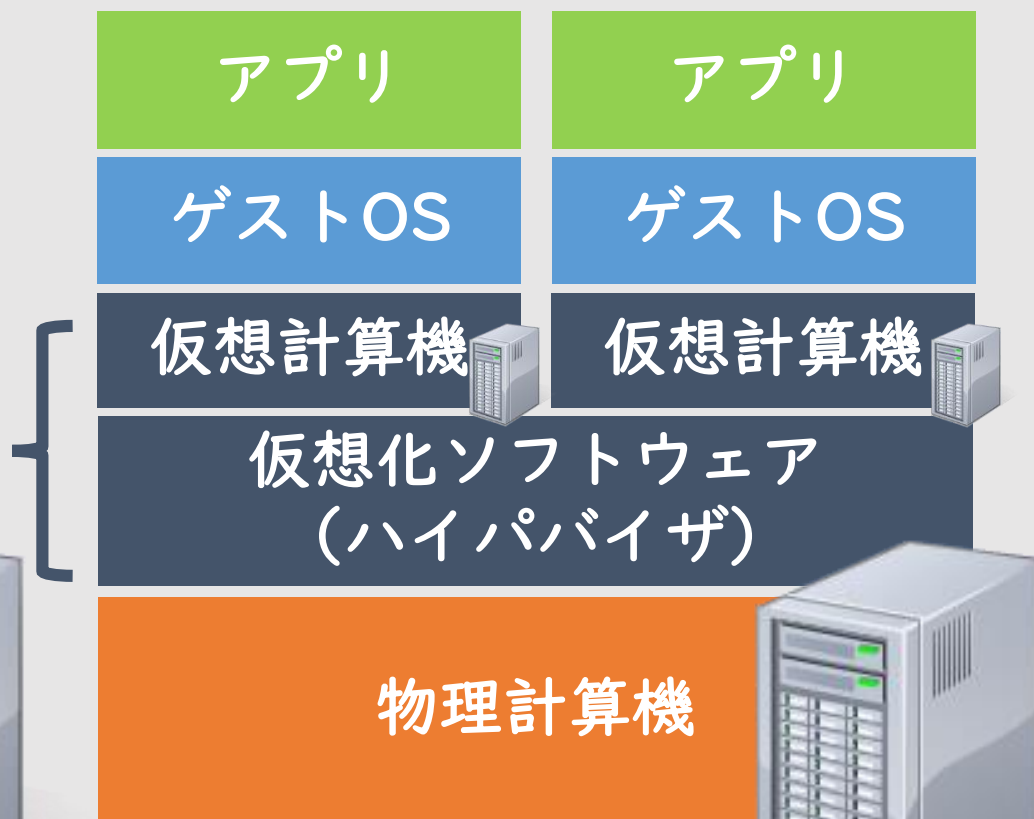
クラウドデータセンタ

仮想計算機 (Virtual Machine)

ソフトウェアで作り出された仮想的な計算機

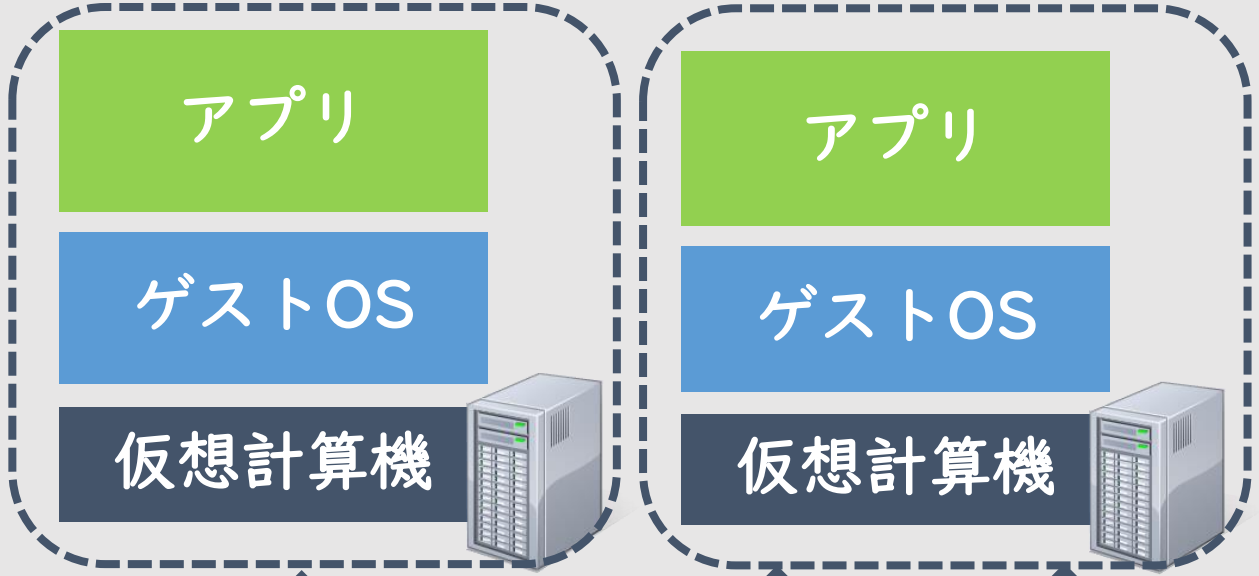
仮想計算機

従来の計算機



仮想計算機の動的な性質

ソフトウェアで動的に生成・破棄が可能



クラウド概論のコンテンツ

- クラウド登場までの歴史
- クラウドの定義と特徴
- クラウドとWeb技術

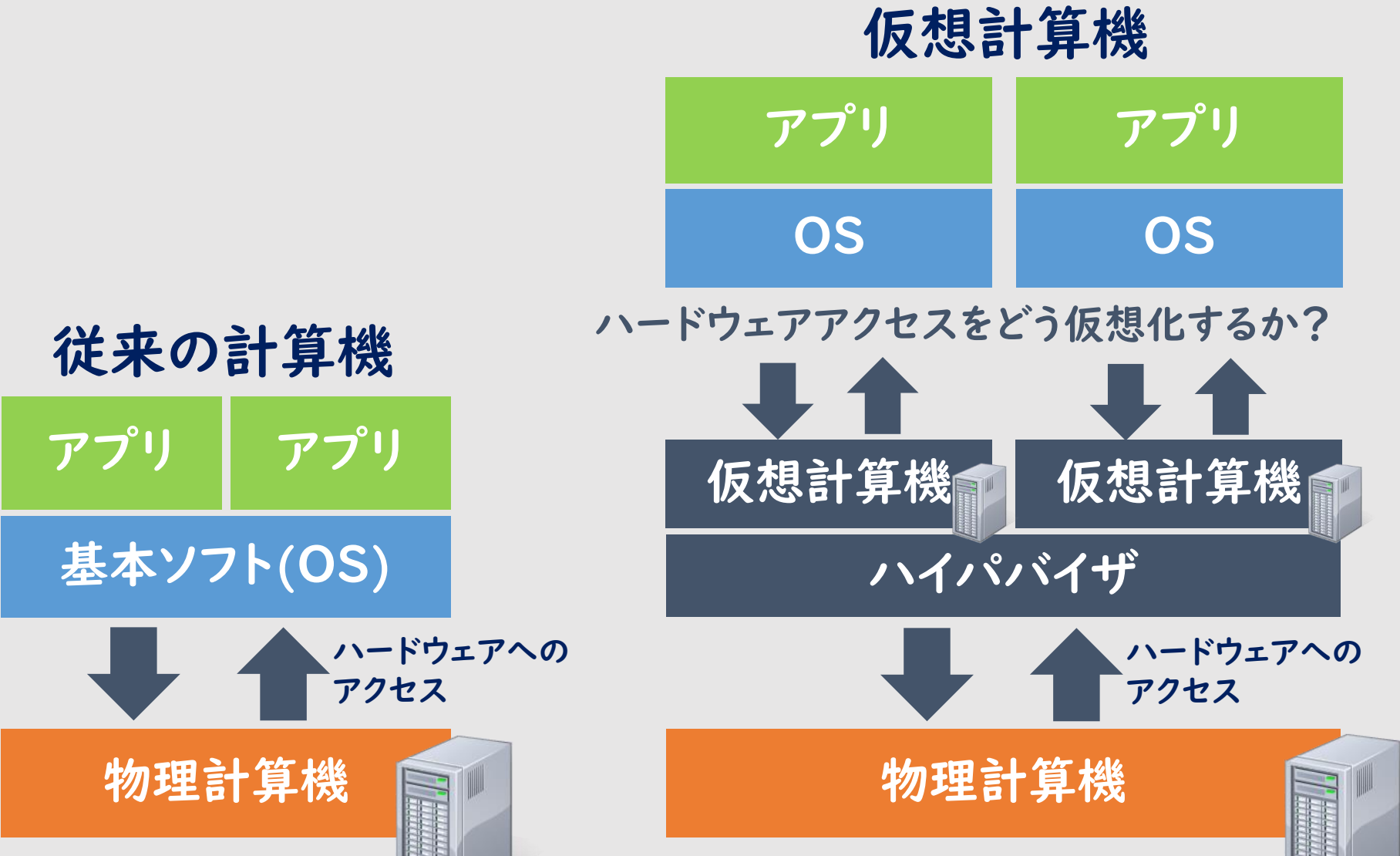
クラウド登場までの歴史



世界初の仮想計算機：
IBM System/360 (1967)

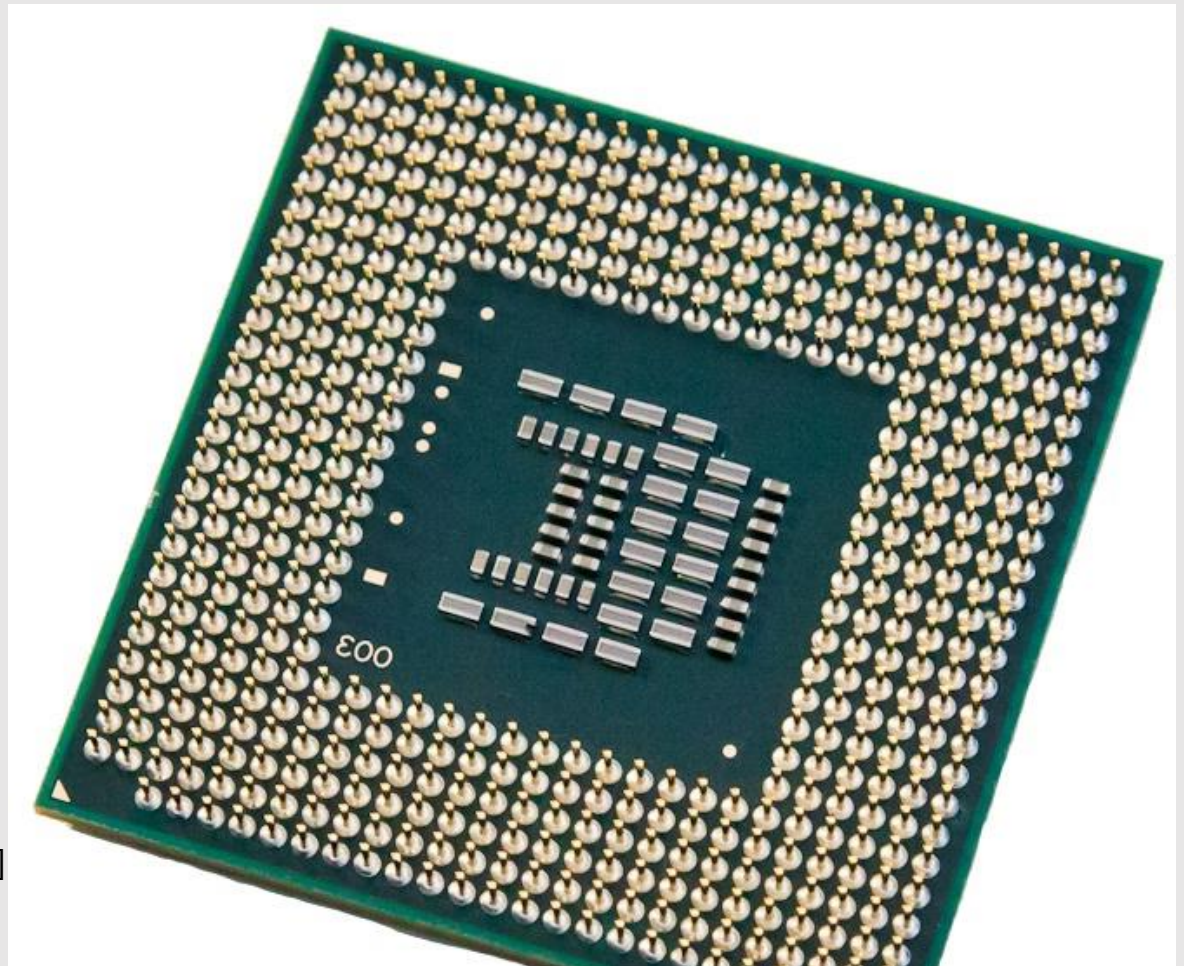
仮想計算機実装の困難さ

OSからのハードウェアアクセスの仮想化



仮想化支援機能の欠如

- 2000年代初頭のIntel系CPUは
- 仮想化機能をハードウェアとして未実装



Xen: Para-virtualization (準仮想化)



Xen and the Art of Virtualization

OSそのものを修正して仮想化に効率よく対応



Xen and the Art of Virtualization

Paul Barham*, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris,
Alex Ho, Rolf Neugebauer†, Ian Pratt, Andrew Warfield

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue, Cambridge, UK, CB3 0FD
(firstname.lastname)@cl.cam.ac.uk

ABSTRACT

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.

This paper presents Xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality. This is achieved by providing an idealized virtual machine abstraction to which operating systems such as Linux, BSD and Windows XP, can be ported with minimal effort.

Our design is targeted at hosting up to 100 virtual machine instances simultaneously on a modern server. The virtualization approach taken by Xen is extremely efficient: we allow operating systems such as Linux and Windows XP to be hosted simultaneously for a negligible performance overhead — at most a few percent compared with the unvirtualized case. We considerably outperform competing commercial and freely available solutions in a range of microbenchmarks and system-wide tests.

Categories and Subject Descriptors

D.4.1 [Operating Systems]: Process Management; D.4.2 [Operating Systems]: Storage Management; D.4.8 [Operating Systems]: Performance

General Terms

Design, Measurement, Performance

Keywords

Virtual Machine Monitors, Hypervisors, Paravirtualization

*Microsoft Research Cambridge, UK

†Intel Research, Cambridge, UK

1. INTRODUCTION

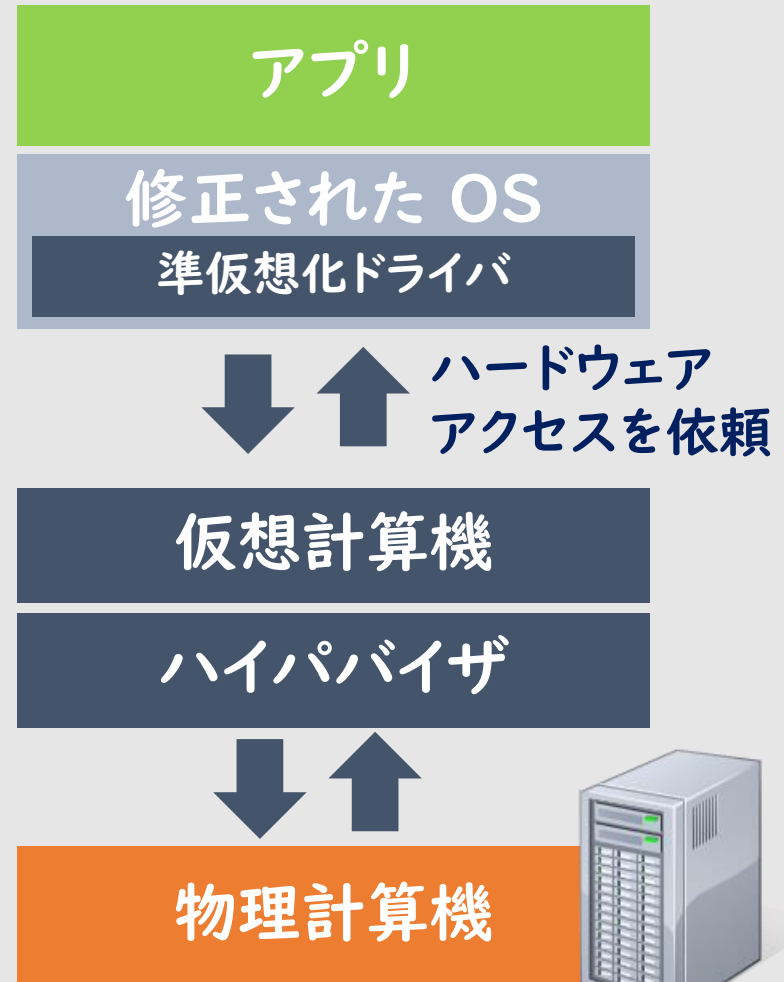
Modern computers are sufficiently powerful to use virtualization to present the illusion of many smaller *virtual machines* (VMs), each running a separate operating system instance. This has led to a resurgence of interest in VM technology. In this paper we present Xen, a high performance resource-managed virtual machine monitor (VMM) which enables applications such as server consolidation [42, 8], co-located hosting facilities [14], distributed web services [43], secure computing platforms [12, 16] and application mobility [26, 37].

Successful partitioning of a machine to support the concurrent execution of multiple operating systems poses several challenges. Firstly, virtual machines must be isolated from one another: it is not acceptable for the execution of one to adversely affect the performance of another. This is particularly true when virtual machines are owned by mutually untrusting users. Secondly, it is necessary to support a variety of different operating systems to accommodate the heterogeneity of popular applications. Thirdly, the performance overhead introduced by virtualization should be small.

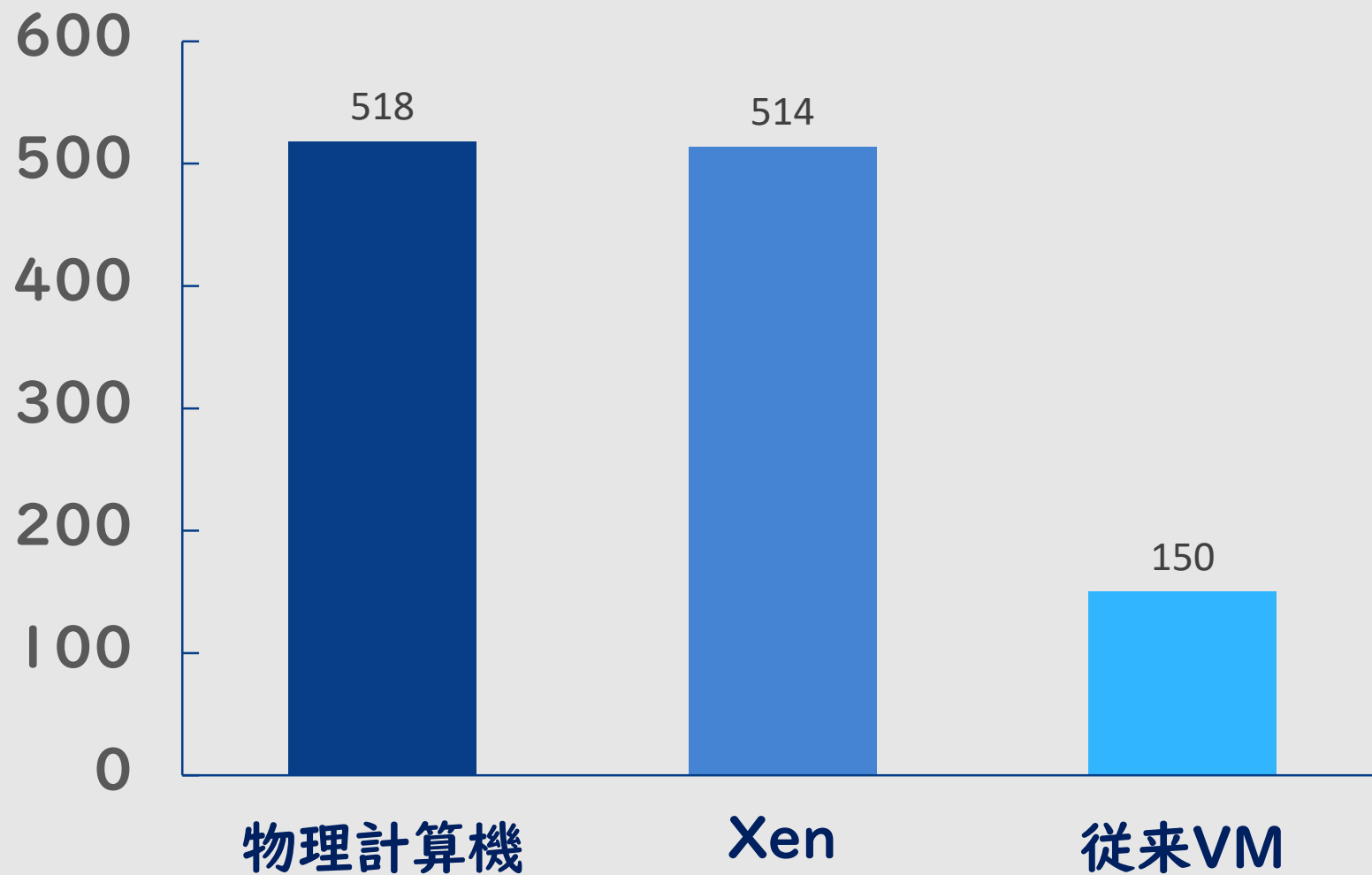
Xen hosts commodity operating systems, albeit with some source modifications. The prototype described and evaluated in this paper can support multiple concurrent instances of our Xen0.linux guest operating system; each instance exports an application binary interface identical to a non-virtualized Linux 2.4. Our port of Windows XP to Xen is not yet complete but is capable of running simple user-space processes. Work is also progressing in porting NetBSD.

Xen enables users to dynamically instantiate an operating system to execute whatever they desire. In the XenServer project [15, 35] we are deploying Xen on standard server hardware at economically strategic locations within ISPs or at Internet exchanges. We perform admission control when starting new virtual machines and expect each VM to pay in some fashion for the resources it requires. We discuss our ideas and approach in this direction elsewhere [21]; this paper focuses on the VMM.

There are a number of ways to build a system to host multiple applications and servers on a shared machine. Perhaps the simplest is to deploy one or more hosts running a standard operating system such as Linux or Windows, and then to allow users to install files and start processes on these commodity operating systems.



準仮想化のパフォーマンス



SPEC WEB99 (score) cited from "Xen and the Art of Virtualization"

Intel Virtualization Technology (Intel VT)の発表

- ハードウェアによる仮想化の支援機能の誕生
 - Xenによる仮想計算機の普及がハードウェアメーカーを動かした

SR-IOV: 入出力デバイスの共有支援

Intel VT-d: 入出力デバイスの仮想化支援

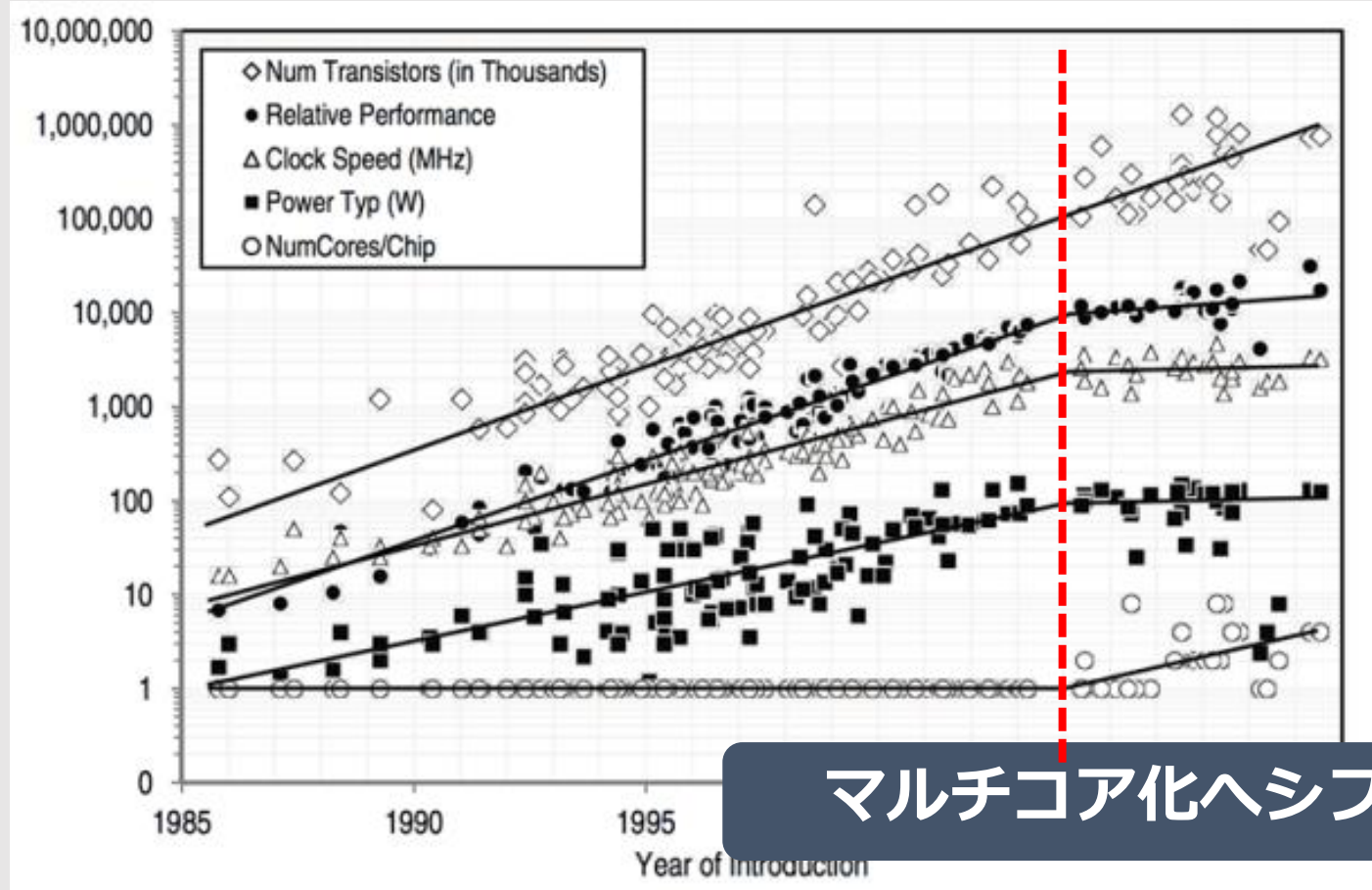
Intel VT-x: CPUの仮想化支援

2005

Intelの仮想化ロードマップ

仮想化技術に対する別の動機

- ムーアの法則はまだ続いているが...



National Research Council. *The Future of Computing Performance: Game Over or Next Level?*. Washington, DC: The National Academies Press, 2011.

需要と供給のミスマッチ

2コアだけ
使いたい



4コア
使いたい

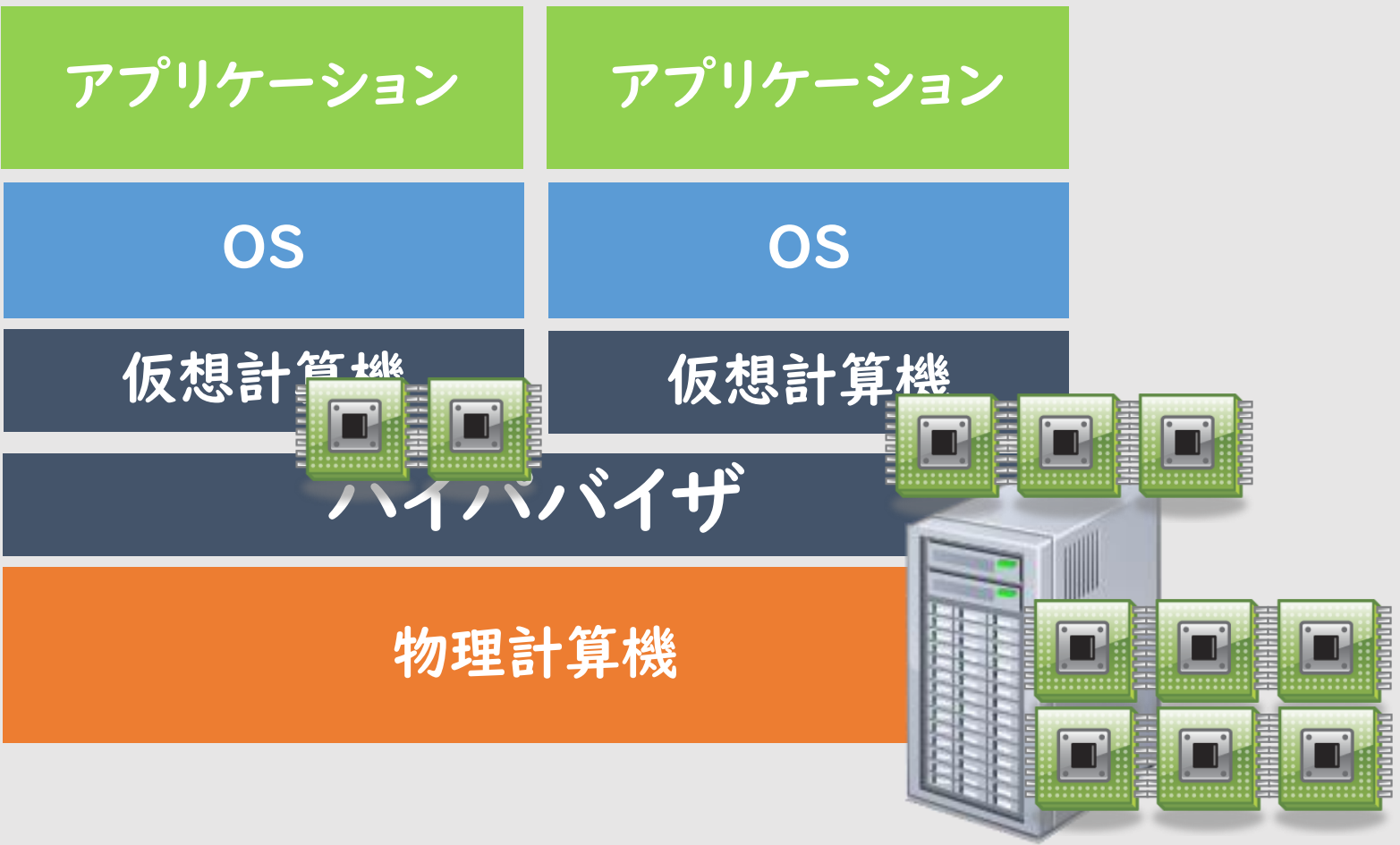


1コアだけで
十分



80コア搭載マシンです。
使ってください!

仮想化による計算機資源の有効な割当



ソフトウェアによる 仮想計算機資源の柔軟な制御

```
# virt-clone -o vm01 -n vm02
```

```
# virsh migrate vm01  
qemu+ssh://host2/system
```

複製

移動

作成

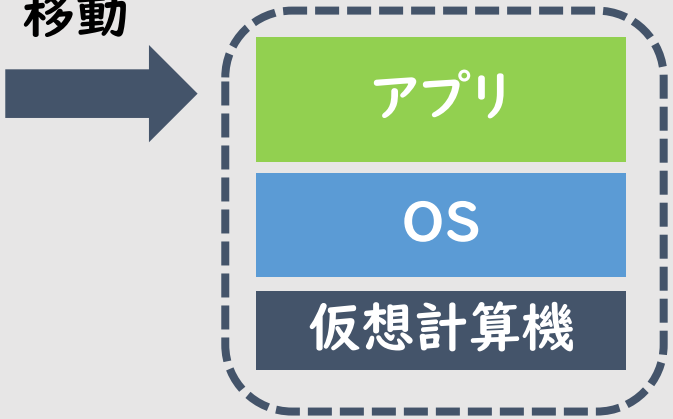
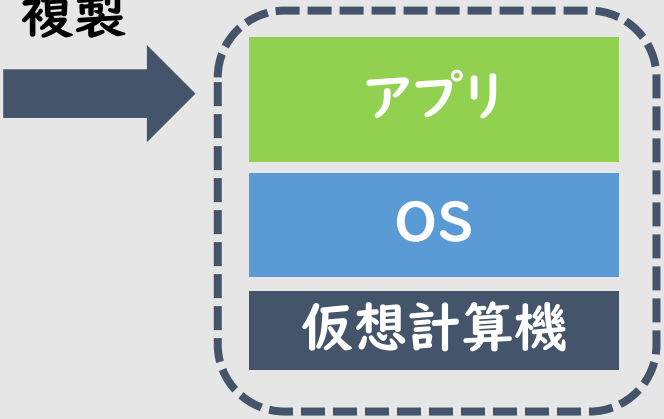
```
# virsh create vm01
```

ハイパバイザ

物理計算機

ハイパバイザ

物理計算機



仮想計算機技術による クラウドデータセンターの実現

ソフトウェアで仮想的に構築された計算機環境



ソフトウェアで仮想的に構築された計算機環境



物理的な計算機上にソフトウェアで動的に環境構築



リソースプール

マルチコアCPU

大規模ストレージ

高速ネットワーク

クラウドデータセンター

代表的クラウドプラットフォーム： Amazon Web Services

- ブラウザから仮想計算機の作成・起動・終了
- 1時間単位の低コスト課金
0.65 ~ 383円/h
- 誰でも簡単に仮想計算機, サービス作成



A screenshot of the Amazon EC2 Management Console. The browser address bar shows the URL "https://console.aws.amazon.com/ec2/home?region=us-east-1#s=Instances". The console interface includes a navigation menu on the left with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORAGE, and NETWORK & SECURITY. The main content area shows a table of instances. One instance named "demo1" is listed with details such as AMI ID, Root Device, Type (t1.micro), State (running), Status Checks (2/2 checks passed), Alarm Status (none), Monitoring (basic), and Security Groups (quicklaunch-1). Below the table, there is a message "No EC2 Instances selected." and a prompt "Select an instance above". The Amazon EC2 logo is visible in the bottom right corner of the console interface.

Amazonに続くクラウドプラットフォーム



Google
Cloud Platform

(2008)



Azure

(2008)



IBM **Cloud**

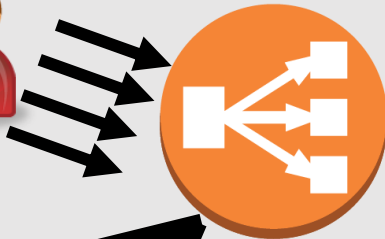
(2014)

クラウドの世界シェア

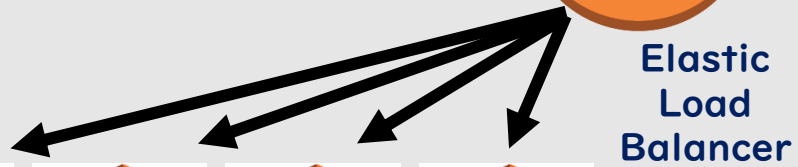
Worldwide cloud infrastructure spending and annual growth
Canalys estimates, full-year 2019

Cloud service provider	Full-year 2019 (US\$ billion)	Full-year 2019 market share	Full-year 2018 (US\$ billion)	Full-year 2018 market share	Annual growth
AWS	34.6	32.3%	25.4	32.7%	36.0%
Microsoft Azure	18.1	16.9%	11.0	14.2%	63.9%
Google Cloud	6.2	5.8%	3.3	4.2%	87.8%
Alibaba Cloud	5.2	4.9%	3.2	4.1%	63.8%
Others	43.0	40.1%	34.9	44.8%	23.3%
Total	107.1	100.0%	77.8	100.0%	37.6%

Dropboxの誕生

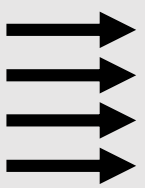


ロードバランサ



Amazon EC2 Amazon EC2 Amazon EC2 Amazon EC2

ブロックサーバ



Elastic Load Balancer



Amazon S3



クラウドの歴史

人気度の動向



注

December 7, 2006 12:05 PM PST

Newsmaker: 'The world needs only five computers'

By Stephen Shankland
Staff Writer, CNET News

[See all Newsmakers »](#)



newsmaker Industry lore likely is wrong to attribute to IBM Chairman Thomas J. Watson the famous misjudgment that there's a world market for five computers.

But Sun Microsystems Chief Technology Officer Greg Papadopoulos thinks the idea will pan out eventually.

"The world needs only five computers," [Papadopoulos said on his blog](#). He then listed seven--Google, eBay, Amazon.com, Microsoft, Yahoo, Salesforce.com, and what he called the Great Computer of China--but let's not split hairs. He was trying to make the point that "there will be, more or less, five hyperscale, pan-global broadband computing services giants."

[Sun caters to pan-global hyperscale customers](#), so it's not surprising that its top technologist sees the world through data center-colored glasses. But the Santa Clara, Calif.-based company's vision has been right on target more than once--notably in forecasting the rise of the Internet.

Papadopoulos detailed his vision in an interview recently with

Related Stories

[Amazon: Utility computing power broker](#)

November 16, 2006

[Sun sees future in computing service providers](#)

October 25, 2006

[Salesforce plans on-](#)

クラウドの定義と特徴

NISTのクラウドの定義

- The NIST Definition of Cloud Computing
 - NIST（米国国立標準技術研究所）が整理したクラウドの定義（2011）
- クラウドの5つの基本要素
 - On-demand self-service
 - Broad network access
 - Resource pooling
 - Rapid elasticity
 - Measured Service

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-145

The NIST Definition of Cloud Computing

Recommendations of the National Institute
of Standards and Technology

Peter Mell
Timothy Grance

クラウドの特徴

必要な時に
すぐ作成



オンデマンド

低価格



セルフサービス

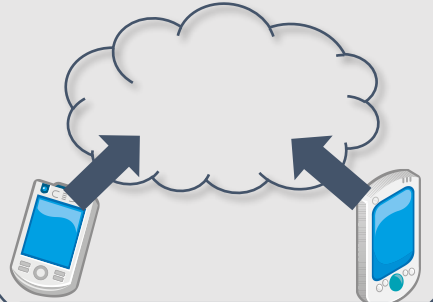
測定サービス

使用分のみ支払い



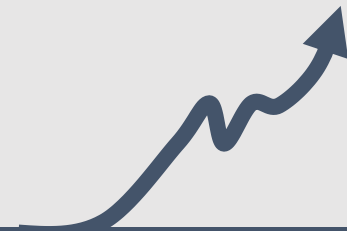
測定サービス

どこからでも
利用可能



ブロードネットワーク

スケールアップ・
ダウンが容易



測定サービス

迅速な拡張

膨大なリソース



リソースプーリング

迅速な拡張

クラウドによる変化

- ITリソースの所有から利用への転換
 - 物理機器の調達・メンテナンスからの解放
- スモールスタート&スケールアップ
 - 低いリスクで事業を開始可能
- サービスの可用性向上
 - 低故障率、地域分散による耐災害性確保
- アプリケーション開発スピードの加速
 - 専用インフラ構築からインフラサービス利用へ
- インフラ構築のソフトウェア化
 - インフラ構築の自動化・プログラマブル化

アプリケーションの開発スピードの加速

- アプリケーションエンジニアとインフラエンジニアの対立



インフラに対する様々な要求 \longleftrightarrow システムに起因する制約

- インフラのサービス化・カタログ化による対立解消



様々なインフラ機能のサービス化

データベースサービス



Amazon RDS



DynamoDB

ストレージ

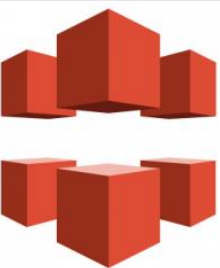


Amazon S3



Amazon Glacier

ネットワーク



CloudFront



Route 53

データ分析/人工知能



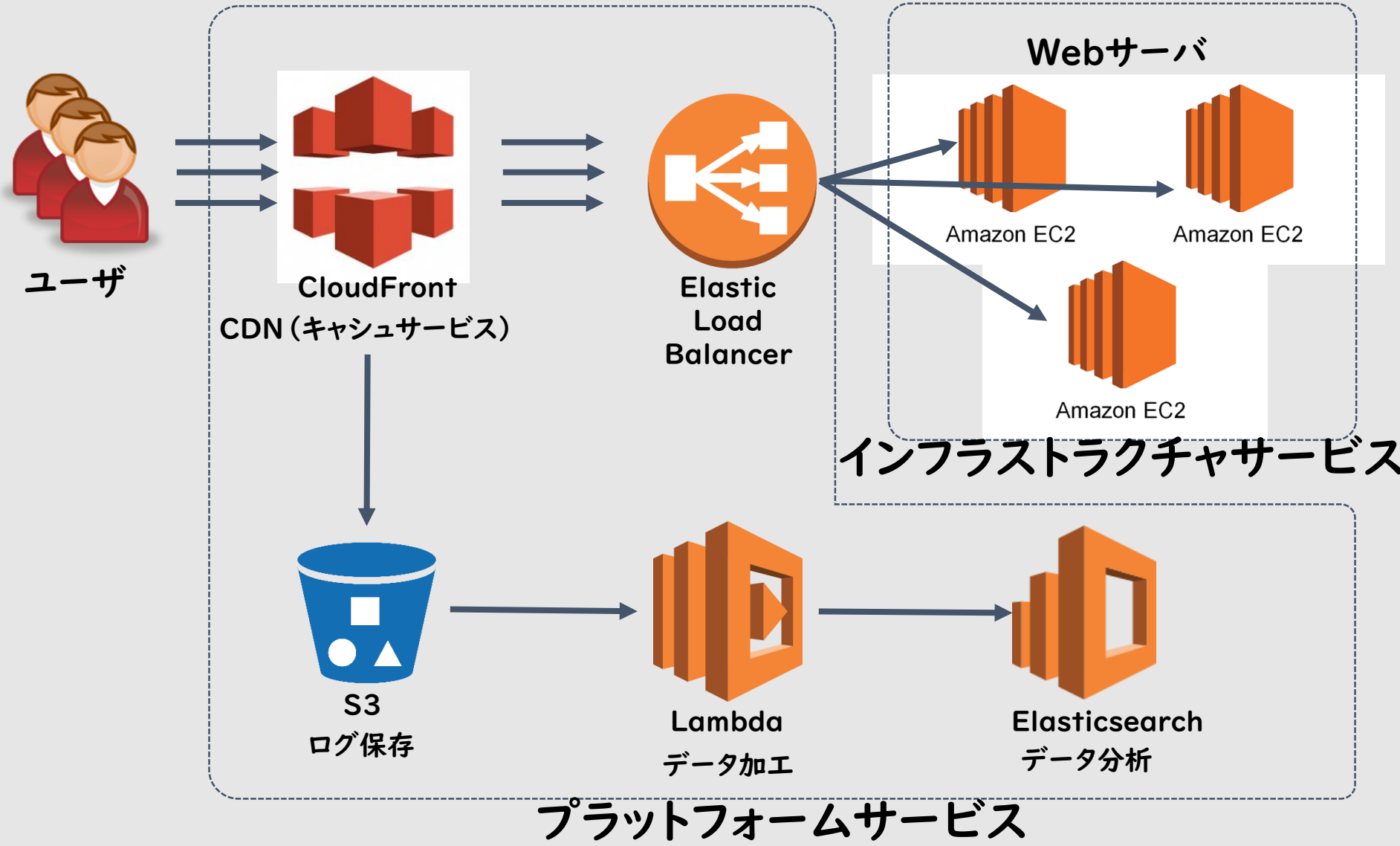
EMR



Map Reduce

単なるインフラストラクチャサービスから
アプリケーション開発のプラットフォームへ

クラウドにおけるシステム開発



クラウドは安いのか？

- 何をコストと思うかで変化
 - 利用料金はそこまで安くない
- オンプレで目に見えにくいコスト
 - 人員確保
 - 設置場所
 - 高可用性運用
 - 各種申請
 - 電力計算

調達の間



ラック間配線は天井？床下？
工事の申請はいつまでに必要？

LAN線の色は？
長さは？
本数は？

合計電力量は？
力率はどの程度で計算する？

この機器の見積もりとったっけ？
妥当な金額まで値切った？
相見積もりをとって比較した？
この見積もりの有効期間は？

機器の搬入経路は？
搬入時の申請は？
台車は何台必要？

このNW機器はどのラック？
この機器を設置するのに必要な
のは何ユニットのスペース？

OSインストールメディアはいくついる？
めっちゃうちゃ寒いけど何時間交代で作業？

開梱作業は誰？
キittingは必要？
ラッキングは誰がやる？

保守作業は誰がやる？
日々異音・異臭のチェックはできる？

床面積あたりの耐荷重は？
設置するサーバの合計重量は？
ストレージ装置は下のほうがよい？

クラウド調達

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Launch Templates
 - Spot Requests
 - Reserved Instances
 - Dedicated Hosts
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs

Launch Instance ▼ Connect Actions ▼

Filter by tags and

Launch Instance
Launch instance from template

<input type="checkbox"/>	Name	Instance ID	Instance Type	Region	Status	Health	Checks	OK
<input type="checkbox"/>		i-0b49c1a0e7e345b99	m4.xlarge					
<input type="checkbox"/>		i-0722408b4a1ba22...	m4.xlarge					
<input type="checkbox"/>		i-dd929a78	m3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-08813facbdb288e51	m3.large	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0e89cb48851adc9e6	m3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0ddb32bdb5330360	c3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0c56ccbbaee24125af	c3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0802ca6c5575d7422	c3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0c3aada50f3710e0e	c3.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-005a38213efe3b773	m4.xlarge	ap-northeast-1a	running	2/2 checks ...	OK	
<input type="checkbox"/>		i-0a66a950c7ce5ac31	r3.2xlarge	ap-northeast-1a	running	2/2 checks ...	OK	

クリックするだけ

Select an instance above

クラウドのサービスモデル

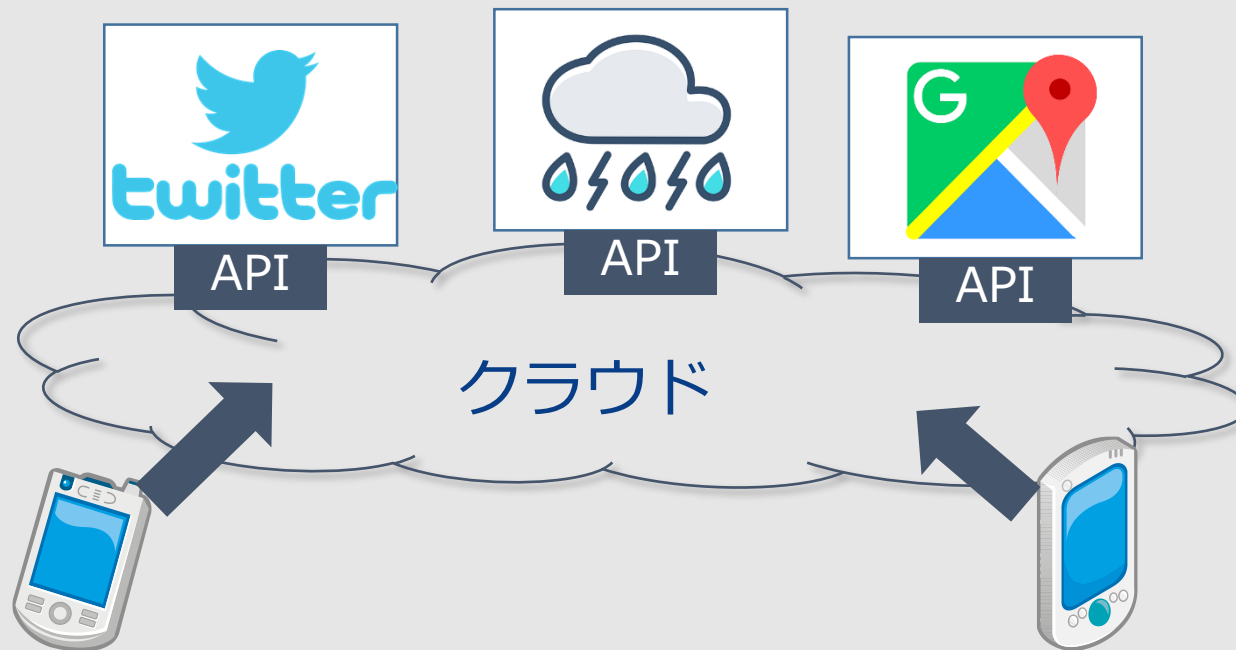
- 3つのサービス提供モデル
- Infrastructure as a Service (IaaS)
 - クラウド上で提供される計算機資源そのものの提供サービス (e.g. Amazon EC2)
- Platform as a Service (PaaS)
 - クラウド上で提供されるソフトウェアプラットフォーム (e.g. Amazon EMR, AzureML)
- Software as a Service (SaaS)
 - クラウド上で実装されたソフトウェアをサービスとして提供 (e.g. Dropbox, Gmail)

Everything as a Service XaaS

クラウドとWeb技術

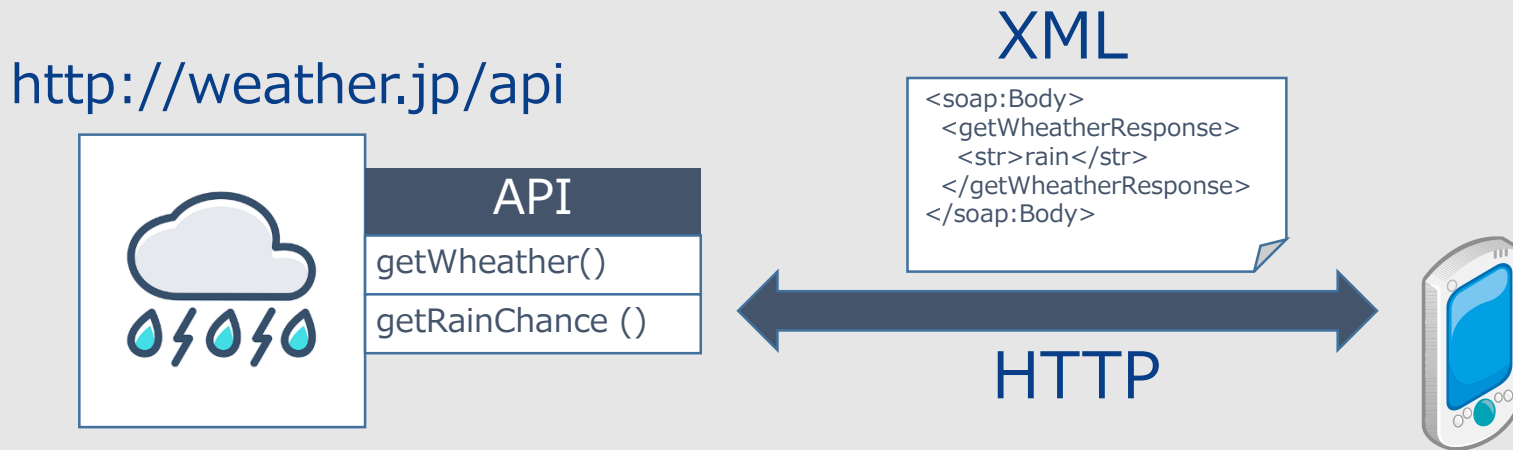
クラウドのWebサービス化

- Webを通じてサービスをAPIとして提供
 - ブロードネットワークアクセス
（どこからでもサービスが利用可能）
 - 分散する異なるサービスを組み合わせ
て新しいサービスを構成（マッシュアップ）



Webサービスのプロトコル

- SOAP
 - Webサービスにオブジェクト指向の概念を導入したもの
 - サービス指向アーキテクチャ



リモートのオブジェクトの
メソッド呼び出しを実現

(ただし、Webアーキテクチャ的に直感的でなく複雑)

Webアーキテクチャの良さ

- URI (Uniform Resource Identifier)
 - シンプルな表記でWeb上の一意のリソースを識別する (<http://weather.jp/xxxx/yyyy>)
 - WebはURIで識別される無数のリソースの集合

<http://weather.jp/rain>

<http://twitter.com/tweets>

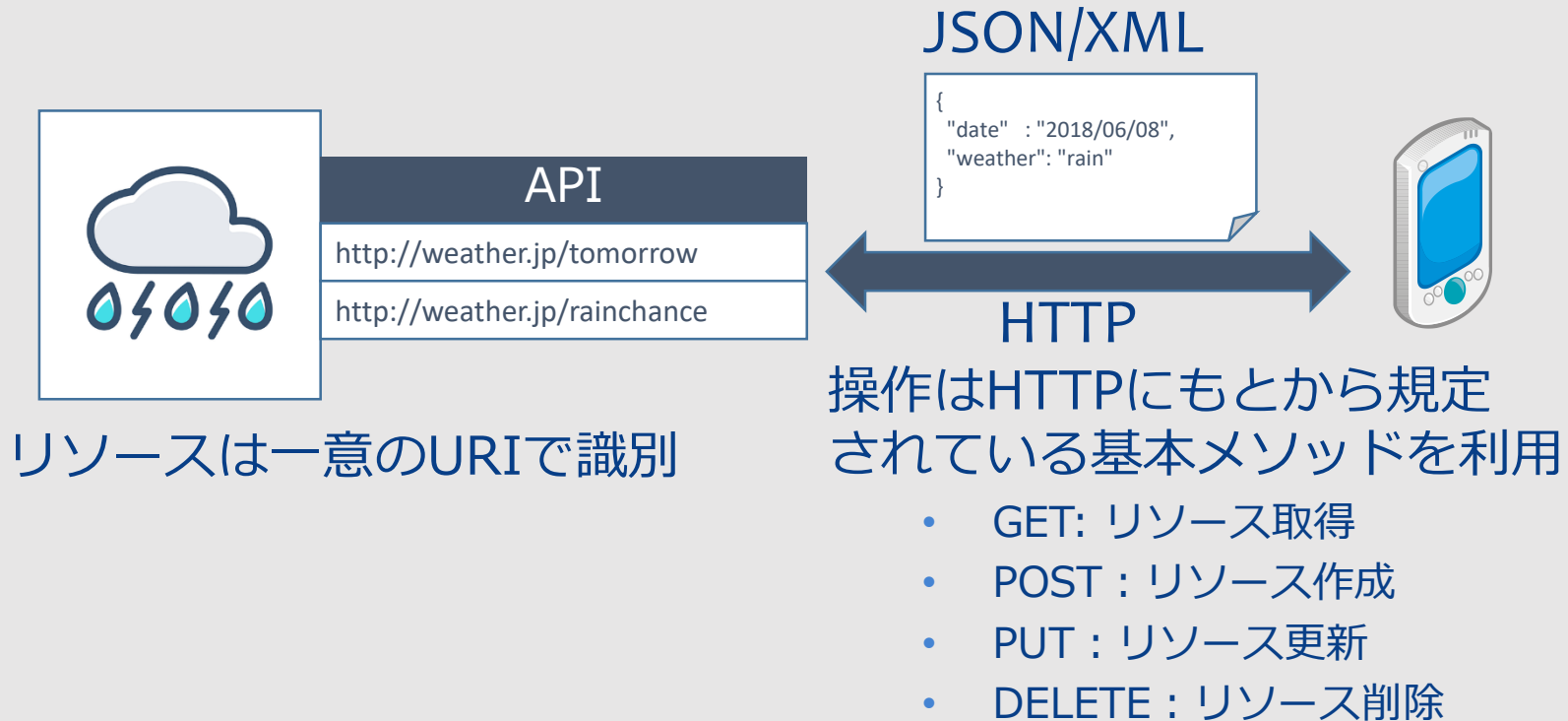
Web

<http://weather.jp/tomorrow>

<http://twitter.com/followers>

RESTful API

- リソース指向アーキテクチャ



HTTPの基本メソッドを使用するので：

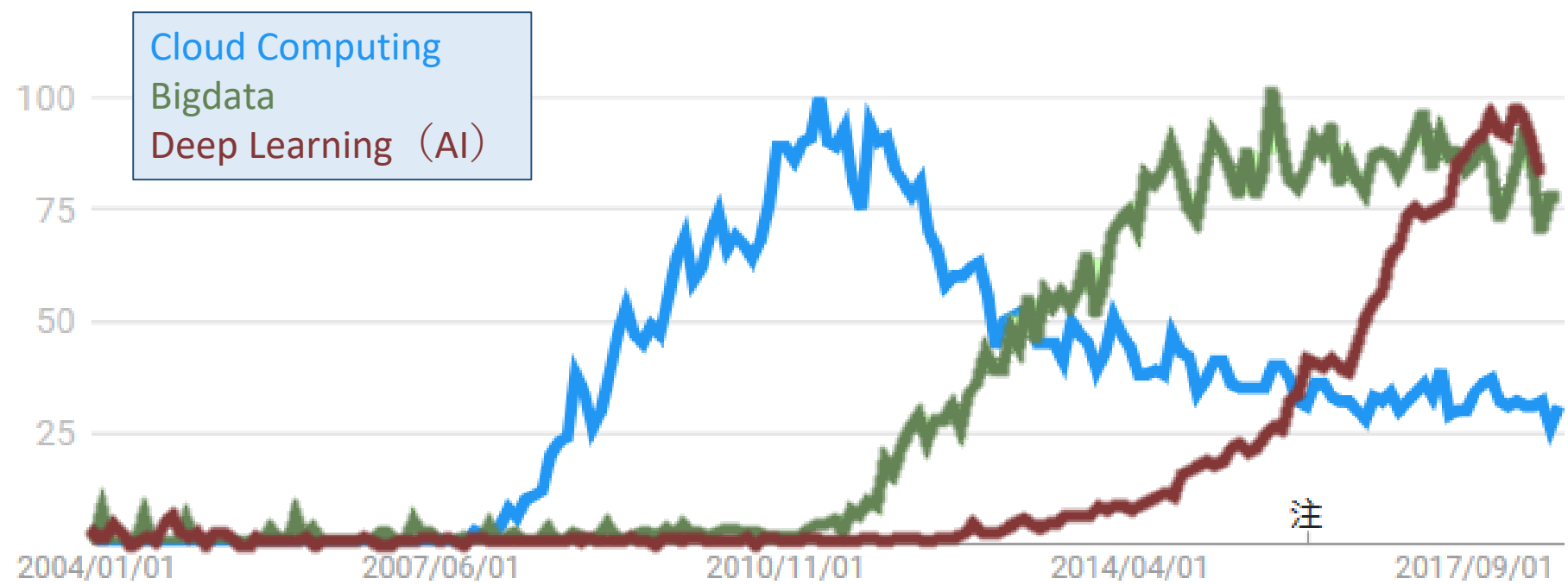
- 全てのサービスで共通の操作
- ブラウザからも動作確認可能

クラウドとは「計算機のソフトウェア化」がもたらした
コンピューティングの進化の形

C → B → A

重ねてみるとよく分かる

人気度の動向



まとめ

- 昨今のAIは「データ」と「計算機」に支えられている
 - Deep Learning
- ビッグデータを扱うためのソフトウェア、ハードウェア基盤が整った
 - Hadoop, 分散処理システム
- 仮想化技術によってクラウドが発展
 - 高性能, 大容量, 大規模な計算機が誰でも使える

課題

- 昨今の「AI」発展の背景を「ビッグデータ」「クラウド」の2つの観点から自分たちの言葉で説明する文章を作成して下さい。