Data Management in the Cloud

Introduction
(Lecture 1)

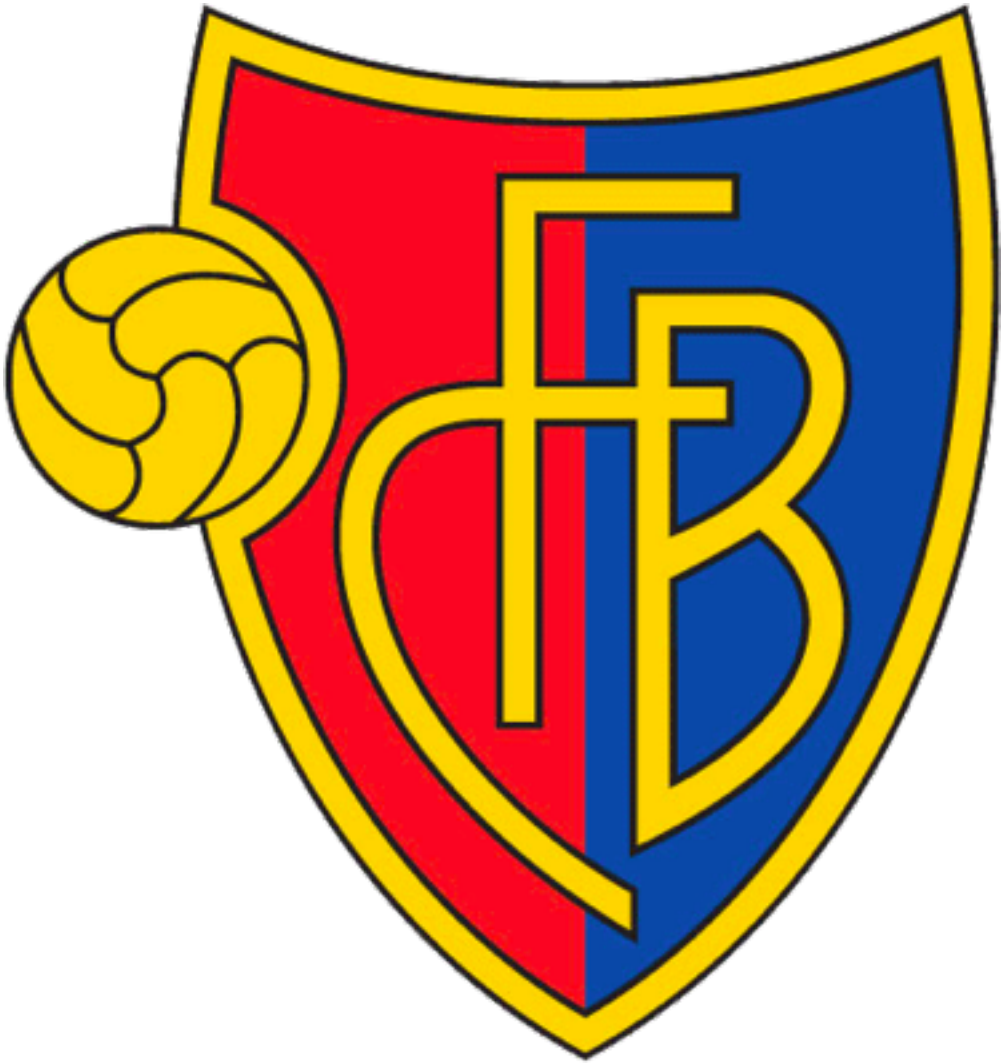
Data Management in the Cloud

LOGISTICS AND ORGANIZATION

Personnel

- Kristin Tufte
  - FAB 115-09
  - Email: tufte@pdx.edu
  - Office hours: right after class
- TA: Hisham Benotman
  - FAB 115
  - Email: benotman@pdx.edu
  - Office hours: TBA

Thanks to Michael Grossniklaus



Course Resources

- Web site
  - Piazza: https://piazza.com/pdx/winter2016/cs410510cloud/home
  - Also in your email and linked off of Kristin's home page
- Lecture note slides
  - online by lecture time
- Literature list
  - Readings associated with most lectures – links on the course web site
- Book - Required
  - NoSQL Distilled: A Brief Guide to the Emerging world of Polyglot Persistence

Course Goals

- **Understand** the basic concepts of cloud computing and cloud data management
- **Learn** how to design data models and algorithms for managing data in the cloud
- **Experiment** with cloud data management systems
- **Work** with cloud computing platforms
- **Compare** and **discuss** results

- Hopefully, have a **good time** doing so!

Planned Course Schedule

I. Introduction and Basics
   - motivation, challenges, concepts, …
   - storage, distributed file systems, and map/reduce, …

II. Data Models and Systems
   - key/value, document, column families, graph, array, …

III. Data Processing Paradigms
   - SCOPE, Pig Latin, Hive, …

IV. Scalable SQL
   - Microsoft SQL Azure, VoltDB, …

V. Advanced and Research Topics
   - SQLShare, benchmarking, …

Course Prerequisites

- Programming skills
  - Java, C#, C/C++
  - algorithms and data structures
  - some distributed systems, e.g. client/server
- Database management systems
  - physical storage
  - query processing
  - optimization

Assignments & Project

- Assignments (45% of Grade)
  - 4 assignments
  - Some individual, some pairs (this is an update)
  - Some question/answer on readings or class discussions, some implementation
- Course Project (50% of Grade)
  - **Part 1**: Data Modeling (Written) (12.5%)
  - **Part 2**: System Profile (Written) (12.5%)
  - **Part 3**: Application Design (Presentation) (12.5%)
  - **Part 4**: Application Implementation (Coding & Presentation) (12.5%)
  - Teams of 4-5 students
- Class Participation (5% of Grade)
  - Daily discussion questions
- Application Implementation presentations will be done during the Finals time slot (no Final Exam)
- Assigned readings – on course web site

Course Project

- The course will be accompanied by a project that is based on a data management scenario
  - **Task 1:** Study question that will take you through a “dry run” of mapping the application to a NoSQL data model and make you think about how to answer some simple queries.
  - **Task 2:** Pick a NoSQL system and compile a systems profile, based on papers and documentation.
  - **Task 3:** Design a management and processing system based on the given application and the previously chosen NoSQL system. This time for real!
  - **Task 4:** Implement a prototype of the graph data management and processing system.
- Teams of 4-5 students

Data Management in the Cloud

INTRODUCTION

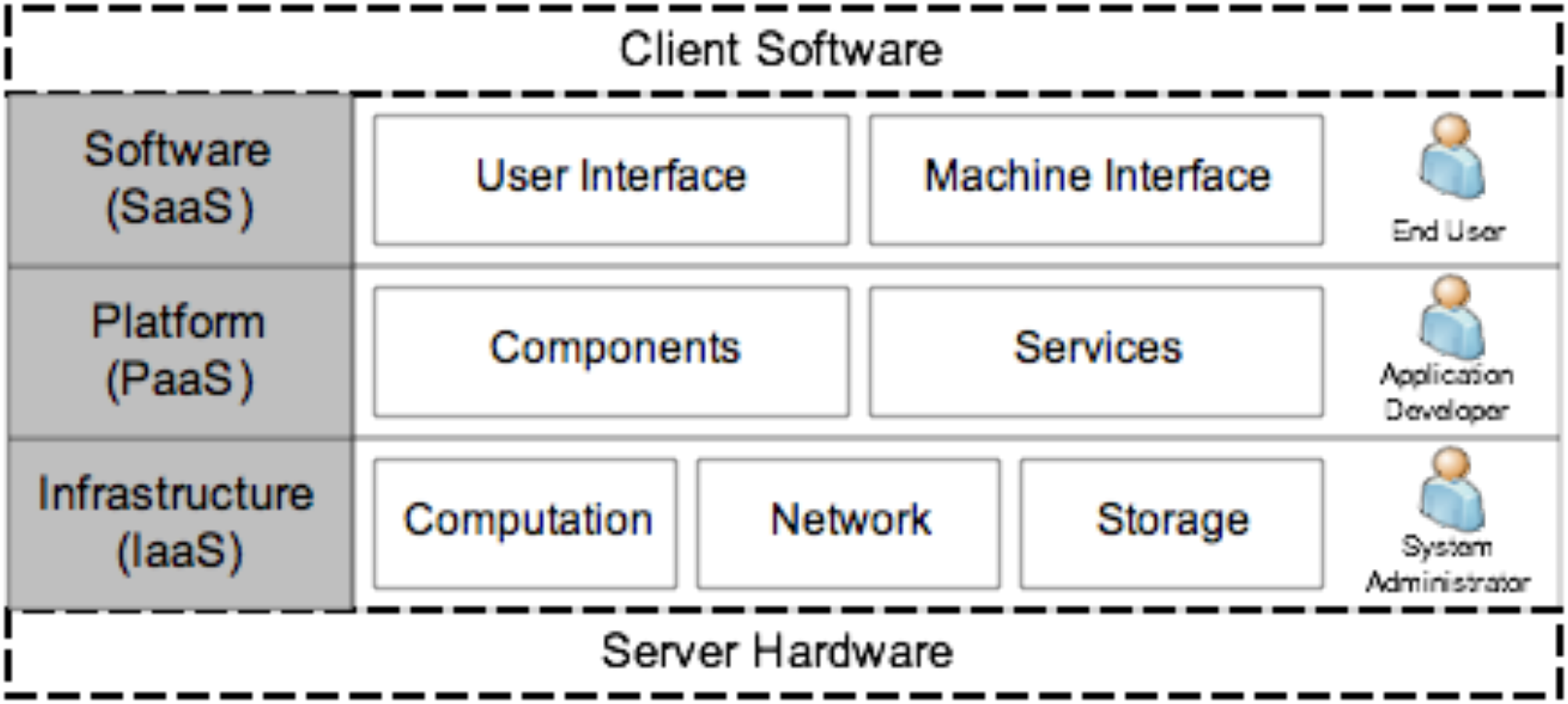
Outline

- Motivation
  - what is cloud computing?
  - what is cloud data management?
- Challenges, opportunities and limitations
  - what makes data management in the cloud difficult?
- New solutions
  - key/value, document, column family, graph, array, and object databases
  - scalable SQL databases

What is Cloud Computing?

- Different definitions for “Cloud Computing” exist
  - How do you define cloud computing?
- Common ground of many definitions
  - processing power, storage and software are **commodities** that are readily available from large infrastructure
  - **service-based view**: “everything as a service (*aaS)”, where only “Software as a Service (SaaS)” has a precise and agreed-upon definition
  - utility computing: **pay-as-you-go** model

Service-Based View on Computing



Source: Wikipedia (http://www.wikipedia.org)

Terminology

- Term **cloud computing** usually refers to both
  - **SaaS**: applications delivered over the Internet as services
  - **The Cloud**: data center hardware and systems software
- Public clouds
  - available in a **pay-as-you-go** manner to the public
  - service being sold is **utility computing**
  - Amazon Web Service, Microsoft Azure, Google AppEngine
- Private clouds
  - internal data centers of businesses or organizations
  - normally not included under **cloud computing**

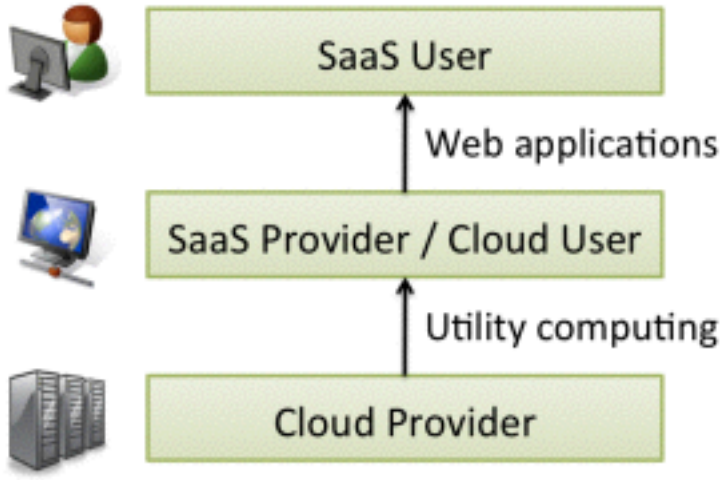
Based on: "Above the Clouds: A Berkeley View of Cloud Computing", RAD Lab, UC Berkeley

Utility Computing

- Illusion of infinite computing resources
  - available on demand
  - no need for users to plan ahead for provisioning
- No up-front cost or commitment by users
  - companies can start small (demand unknown in advance)
  - increase resources only when there is an increase in need (demand varies with time)
- Pay for use on short-term basis as needed
  - processors by the hour and storage by the day
  - release them as needed, reward conservation
- "Cost associativity"
  - 1000 EC2 machines for 1 hour = 1 EC2 machine for 1000 hours

Based on: "Above the Clouds: A Berkeley View of Cloud Computing", RAD Lab, UC Berkeley

Cloud Computing Users and Providers



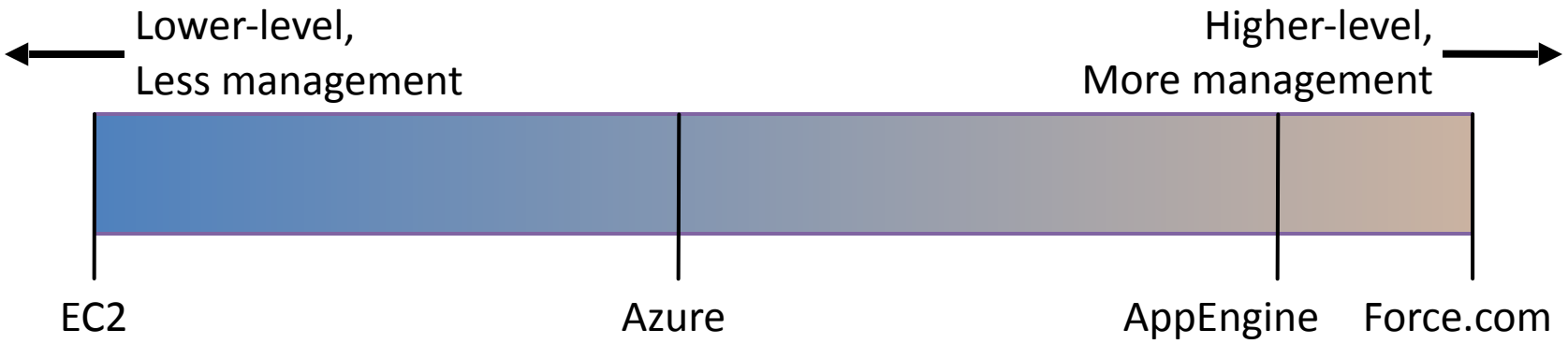
Picture credit: "Above the Clouds: A Berkeley View of Cloud Computing", RAD Lab, UC Berkeley

Virtualization

- Virtual resources abstract from physical resources
  - hardware platform, software, memory, storage, network
  - fine-granular, lightweight, flexible and dynamic
- Relevance to cloud computing
  - centralize and ease administrative tasks
  - improve scalability and work loads
  - increase stability and fault-tolerance
  - provide standardized, homogenous computing platform through hardware virtualization, i.e. **virtual machines**

Spectrum of Virtualization

- Computation virtualization
  - Instruction set VM (Amazon EC2, 3Tera)
  - Byte-code VM (Microsoft Azure)
  - Framework VM (Google AppEngine, Force.com)
- Storage virtualization
- Network virtualization



Slide Credit: RAD Lab, UC Berkeley

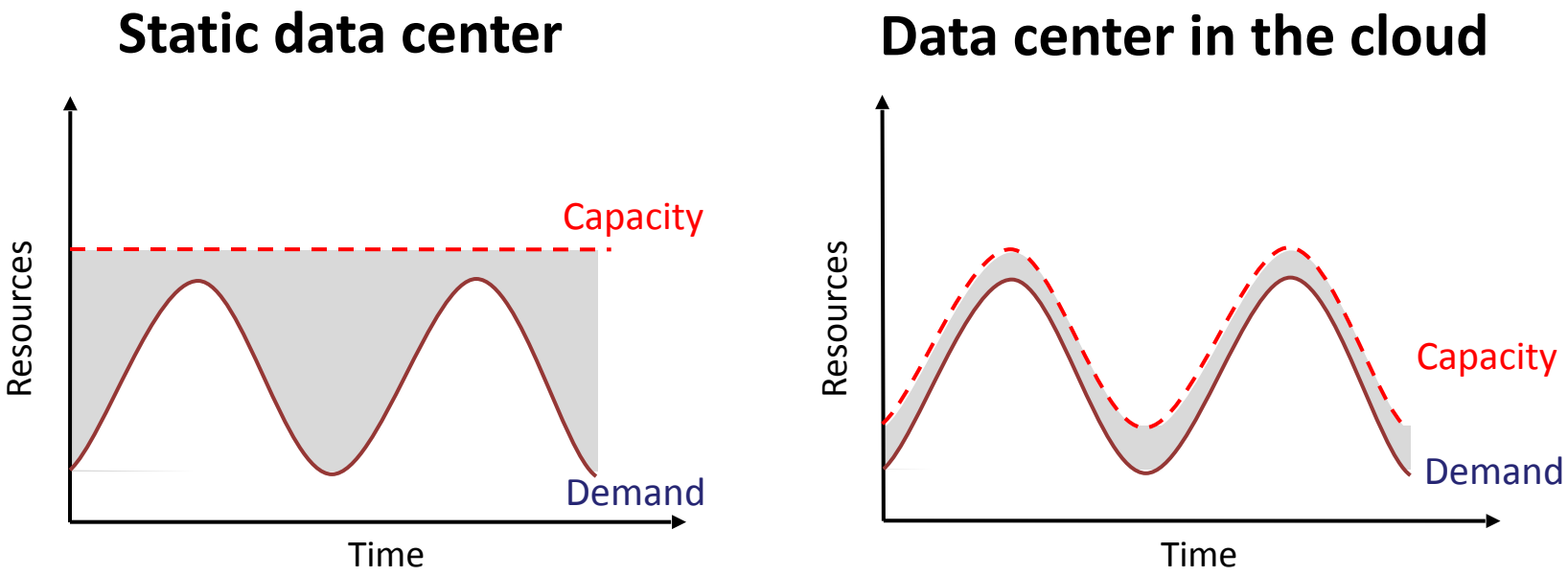
Table 4: Examples of Cloud Computing vendors and how each provides virtualized resources (computation, storage, networking) and ensures scalability and high availability of the resources.

	Amazon Web Services	Microsoft Azure	Google AppEngine
Computation model (VM)	• x86 Instruction Set Architecture (ISA) via Xen VM • Computation elasticity allows scalability, but developer must build the machinery, or third party VAR such as RightScale must provide it	• Microsoft Common Language Runtime (CLR) VM; common intermediate form executed in managed environment • Machines are provisioned based on declarative descriptions (e.g. which "roles" can be replicated); automatic load balancing	• Predefined application structure and framework; programmer-provided "handlers" written in Python, all persistent state stored in MegaStore (outside Python code) • Automatic scaling up and down of computation and storage; network and server failover; all consistent with 3-tier Web app structure
Storage model	• Range of models from block store (EBS) to augmented key/blob store (SimpleDB) • Automatic scaling varies from no scaling or sharing (EBS) to fully automatic (SimpleDB, S3), depending on which model used • Consistency guarantees vary widely depending on which model used • APIs vary from standardized (EBS) to proprietary	• SQL Data Services (restricted view of SQL Server) • Azure storage service	•MegaStore/BigTable
Networking model	• Declarative specification of IP-level topology; internal placement details concealed • Security Groups enable restricting which nodes may communicate • Availability zones provide abstraction of independent network failure • Elastic IP addresses provide persistently routable network name	• Automatic based on programmer's declarative descriptions of app components (roles)	• Fixed topology to accommodate 3-tier Web app structure • Scaling up and down is automatic and programmer-invisible

Table credit: "Above the Clouds: A Berkeley View of Cloud Computing", RAD Lab, UC Berkeley

Economics of Cloud Users

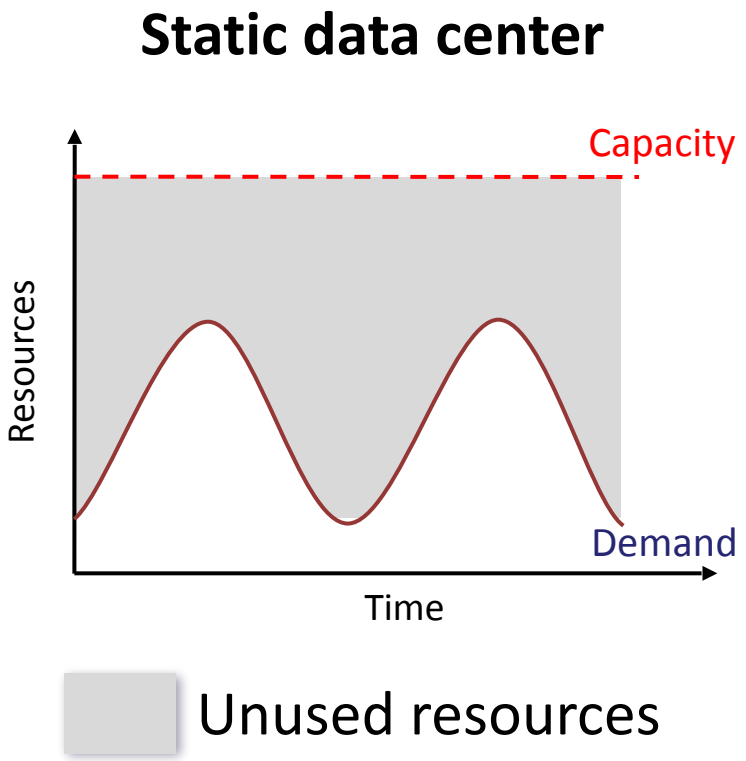
- Pay by use instead of provisioning for peak



Slide Credit: RAD Lab, UC Berkeley

Economics of Cloud Users

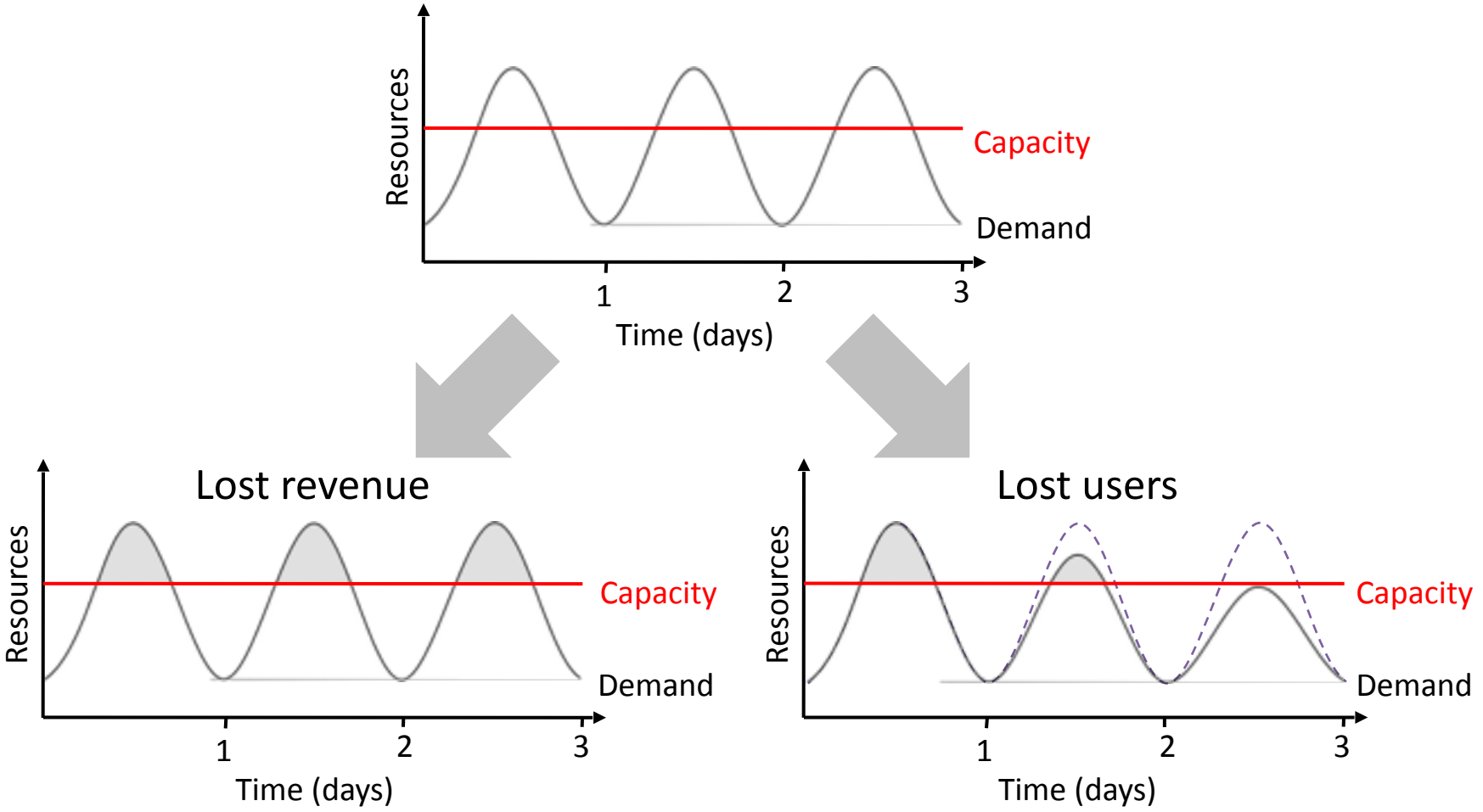
- Risk of over-provisioning: underutilization



Slide Credit: RAD Lab, UC Berkeley

Economics of Cloud Users

- Heavy penalty for under-provisioning



Slide Credit: RAD Lab, UC Berkeley

Economics of Cloud Providers

Resource	Cost in Medium Data Center	Cost in Very Large Data Center	Ratio
Network	$95/Mbps/month	$13/Mbps/month	7.1x
Storage	$2.20/GB/month	$0.40/GB/month	5.7x
Administration	≈140 servers/admin	>1000 servers/admin	7.1x

Source: James Hamilton (http://perspectives.mvdirona.com)

- Cloud computing is 5-7x cheaper than traditional in-house computing
- Power/cooling costs: approx double cost of storage, CPU, network
- Added benefits (to cloud providers)
  - utilize off-peak capacity (Amazon)
  - sell .NET tools (Microsoft)
  - reuse existing infrastructure (Google)

Slide Credit: RAD Lab, UC Berkeley

What is Cloud Data Management?

- Data management applications are potential candidates for deployment in the cloud
  - **industry:** enterprise database system have significant up-front cost that includes both hardware and software costs
  - **academia:** manage, process and share mass-produced data in the cloud
- Many “Cloud Killer Apps” are in fact data-intensive
  - Batch Processing as with map/reduce
  - Online Transaction Processing (OLTP) as in automated business applications
  - Online Analytical Processing (OLAP) as in data mining or machine learning

Scientific Data Management Applications

- Old model
  - "Query the world"
  - data acquisition coupled to a specific hypothesis
- New model
  - "Download the world"
  - data acquired en masse, in support of many hypotheses
- E-science examples
  - astronomy: high-resolution, high-frequency sky surveys, …
  - oceanography: high-resolution models, cheap sensors, satellites, …
  - biology: lab automation, high-throughput sequencing, ...

Slide Credit: Bill Howe, U Washington

Scaling Databases

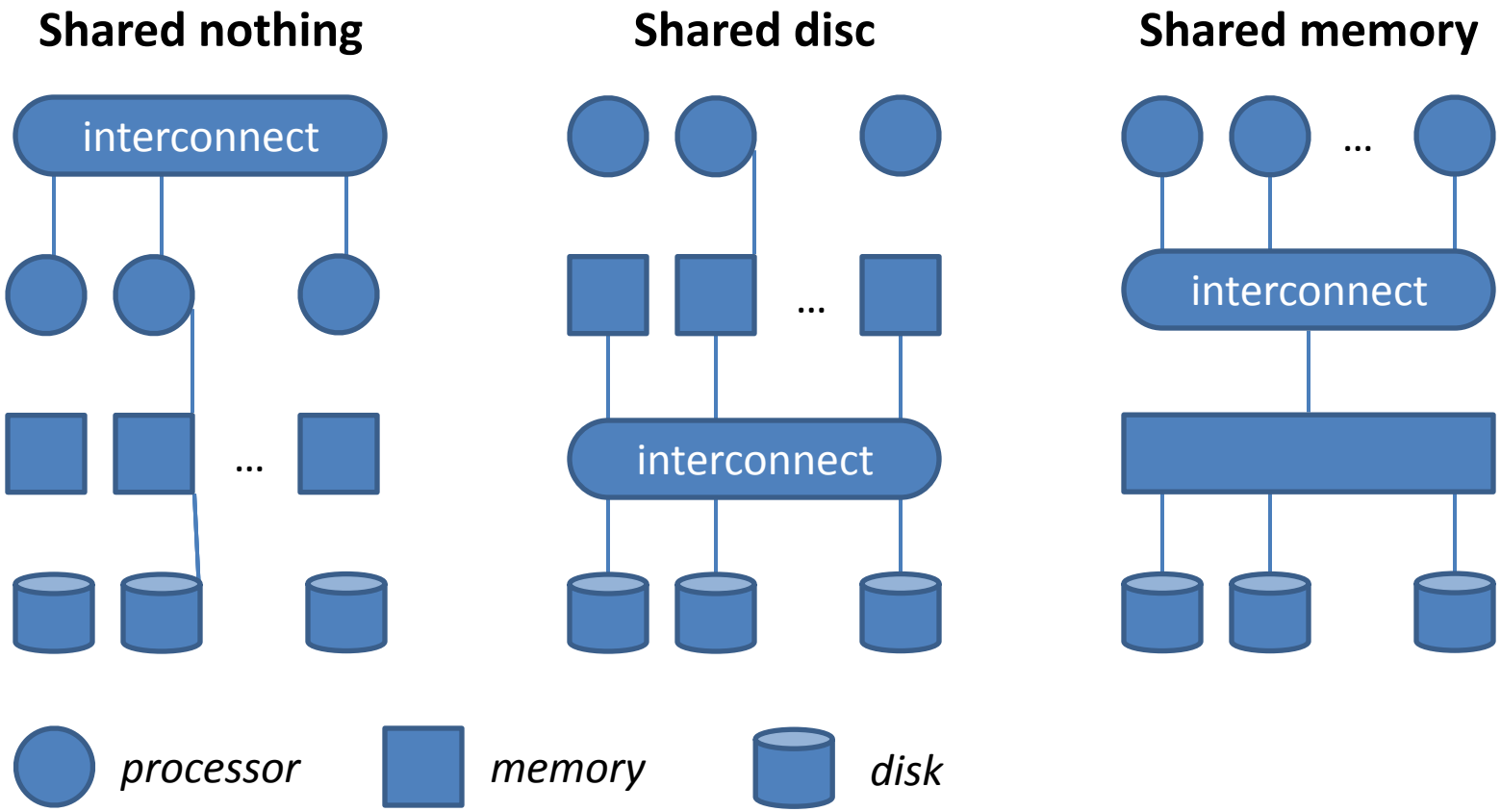
- Flavors of database scalability
  - lots of (small) transactions
  - lots of copies of the data
  - lots of processors running on a single query (compute intensive tasks)
  - extremely large data set for one query (data intensive tasks)
- Data replication
  - move data to where it is needed
  - managed replication for availability and reliability

Revisit Cloud Characteristics

- Compute power is elastic, but only if workload is parallelizable
  - transactional database management systems do not typically use a shared-nothing architecture
  - shared-nothing is a good match for analytical data management
  - some things parallelize well (i.e. sum), some do not (i.e. median)
  - Think about: Google gmail, Amazon web site – easy? Difficult?
  - Google App Engine – API forces ability to run in shared nothing
- Scalability
  - **in the past**: out-of-core, works even if data does not fit in main memory
  - **in the present**: exploits thousands of (cheap) nodes in parallel

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Parallel Database Architectures



Source: D. DeWitt and J. Gray: "Parallel Database Systems: The Future of High Performance Database Processing", CACM 36(6), pp. 85-98, 1992.

Revisit Cloud Characteristics

- Data is stored at an untrusted host
  - there are risks with respect to privacy and security in storing transactional data on an untrusted host
  - particularly sensitive data can be left out of analysis or anonymized
  - sharing and enabling access is often precisely the goal of using the cloud for scientific data sets
  - where exactly is your data? and what are that country's laws?

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Revisit Cloud Characteristics

- Data is replicated, often across large geographic distances
  - it is hard to maintain ACID guarantees in the presence of large-scale replication
  - full ACID guarantees are typically not required in analytical applications
- Virtualizing large data collections is challenging
  - data loading takes more time than starting a VM
  - storage cost vs. bandwidth cost
  - online vs. offline replication

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Challenges

- Trade-off between functionality and operational cost
  - restricted interface, minimalist query language, limited consistency guarantees
  - pushes more programming burden on developers
  - enables predictable services and service level agreements
- Manageability
  - limited human intervention, high-variance workloads, and a variety of shared infrastructures
  - need for self-managing and adaptive database techniques

Based on: "The Claremont Report on Database Research", 2008

Challenges

- Scalability
  - today's SQL databases cannot scale to the thousands of nodes deployed in the cloud context
  - hard to support multiple, distributed updaters to the same data set
  - hard to replicate huge data sets for availability, due to capacity (storage, network bandwidth, …)
  - **storage:** different transactional implementation techniques, different storage semantics, or both
  - **query processing and optimization:** limitations on either the plan space or the search will be required
  - **programmability:** express programs in the cloud

Based on: "The Claremont Report on Database Research", 2008

Challenges

- Data privacy and security
  - protect from other users and cloud providers
  - specifically target usage scenarios in the cloud with practical incentives for providers and customers
- New applications: "*mash up*" interesting data sets
  - expect services pre-loaded with large data sets, stock prices, web crawls, scientific data
  - data sets from private or public domain
  - might give rise to federated cloud architectures

Based on: "The Claremont Report on Database Research", 2008

Transactional Data Management – Cloud or not?

- Transactional Data Management
  - Banking, airline reservation, e-commerce, etc…
  - Require ACID, write-intensive
- Features
  - Do not typically use shared-nothing architectures (changing a bit)
  - Hard to maintain ACID guarantees in the face of data replication over large geographic distances
  - There are risks in storing transactional data on an untrusted host
- Conclusion: not appropriate for the cloud

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Analytical Data Management – Cloud or not?

- Analytical Data Management
  - Query data from a data store for planning, problem solving, decision support
  - Large scale
  - Read-mostly
- Features
  - Shared-nothing architecture is a good match for analytical data management (Teradata, Greenplum, Vertica…)
  - ACID guarantees typically not needed
  - Particularly sensitive data left out of analysis
- Conclusion: appropriate for the cloud

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Cloud DBMS Wish List

- Efficiency
- Fault tolerance (query restart not required, commodity hw)
- Heterogeneous environment (performance of compute nodes not consistent)
- Operate on encrypted data
- Interface with business intelligence products

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Option 1: MapReduce-like software

- Fault tolerance (yes, commodity hw)
- Heterogeneous environment (yes, by design)
- Operate on encrypted data (no)
- Interface with business intelligence products (no, not SQL-compliant, no standard)
- Efficiency (up for debate)
  - Questionable results in the MapReduce paper
  - Absence of a loading phase (no indices, materialized views)

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Option 2: Shared-Nothing Parallel Database

- Interface with business intelligence products (yes, by design)
- Efficiency (yes)
- Fault tolerance (no - query restart required)
- Heterogeneous environment (no)
- Operate on encrypted data (no)

Based on: "Data Management in the Cloud: Limitations and Opportunities", IEEE, 2009.

Why NoSQL?

- Value of relational databases
  - Persistent data
  - Concurrency/transactions
  - Integration
  - (Mostly) Standard Model
- Impedance Mismatch
- Application vs. Integration databases

Based on: "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", 2013

Attack of the Clusters

- Data growth (links, social networks, logs, users)
- Need to scale to accommodate growth
- Traditional RDBMS (Oracle / Microsoft SQL Server) – shared disk – don't scale well
- "technical issues are exacerbated by licensing costs"
  - Google, Amazon influential
- "The interesting thing about Cloud Computing is that we've redefined Cloud Computing to include everything that we already do… I don't' understand what we would do differently in the light of Cloud Computing other than change the wording of some of our ads." – Larry Ellison

Based on: "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", 2013

Emergence of NoSQL

- No strong definition, but…
  - Do not use SQL
  - Typically open-source
  - Typically oriented towards clusters (but not all)
  - Operate without a schema
- Various types (in order of complexity)
  - Key-value stores
  - Document Stores
  - Extensible Record Stores

Based on: "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", 2013

What were we talking about?

- Cloud Computing
  - Utility Computing
  - Virtualization
  - Economics (pay as you go)
- Data management in the cloud
  - Cloud characteristics (elasticity if parallelizable, untrusted host, large distances)
  - Transactional vs. Analytical
  - Wish List
  - Map Reduce vs. Shared-Nothing -> Hybrid
- DB vs. NoSQL in two lines…
  - Database: complex / concurrent
  - NoSQL: simple / scalable

Discussion Question

Pick an online application and discuss its characteristics.

Examples: Twitter, Snapchat, gmail, FaceBook, Minecraft, Healthcare.gov, Dropbox, Flickr, Instagram, Ebay, Yelp, TripAdvisor, Zillow, E*TRADE, iTunes, online banking, Amazon

Three key aspects of the _____ application:

1.

2.

3.

References

- M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, M. Zaharia: **Above the Clouds: A Berkeley View of Cloud Computing.** *Tech. Rep. No. UCB/ EECS-2009-28, 2009.*
- D. J. Abadi: **Data Management in the Cloud: Limitations and Opportunities**. *IEEE Data Eng. Bull. 32(1), pp. 3—12, 2009.*
- R. Agrawal, A. Ailamaki, P. A. Bernstein, E. A. Brewer, M. J. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. J. Franklin, H. Garcia- Molina, J. Gehrke, L. Gruenwald, L. M. Haas, A. Y. Halevy, J. M. Hellerstein, Y. E. Ioannidis, H. F. Korth, D. Kossmann, S. Madden, R. Magoulas, B. Chin Ooi, T. O'Reilly, R. Ramakrishnan, S. Sarawagi, M. Stonebraker, A. S. Szalay, G. Weikum: **The Claremont Report on Database Research.** 2008.
- P. Sadalage, M. Fowler. NoSQL Distilled: **A Brief Guide to the Emerging World of Polyglot Persistence.** 2013