CSE 544
Principles of Database Management Systems

Magdalena Balazinska

Fall 2009

Lecture 15 – Databases as a Service

References

- Amazon SimpleDB, RDS, Elastic MapReduce Websites
  - Part of Amazon Web services
- Google App Engine Datastore Website
  - Part of the Google App Engine
- Microsoft SQL Azure
  - Part of the Azure platform
- Very dynamic space! Need to check docs regularly!

Cloud Computing

- A definition
  - “Style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet”
- Basic idea
  - Developer focuses on application logic
  - Infrastructure and data hosted by someone else in their “cloud”
  - Hence all operations tasks handled by cloud service provider
- Some history
  - "computation may someday be organized as a public utility” (John McCarthy – 1960)
  - 1999, Infrastructure as a Service
  - Early 2000s Web services
  - 2006, Amazon Web Services
  - And now it’s a craze!

Levels of Service

- Infrastructure as a Service (IaaS)
  - Example Amazon EC2
- Platform as a Service (PaaS)
  - Example Microsoft Azure, Google App Engine
- Software as a Service (SaaS)
  - Example Google Docs

How About Data Management as a Service?

- Running a DBMS is challenging
  - Need to hire a skilled database administrator (DBA)
  - Need to provision machines (hardware, software, configuration)
  - Problems:
    - If business picks up, may need to scale quickly
    - Workload varies over time
- Solution: Use a DBMS service
  - All machines are hosted in service provider's data centers
  - Data resides in those data centers
  - Pay-per-use policy
  - Elastic scalability
  - No administration!

Basic Features for Data Management as a Service

- Data storage and query capabilities
- Operations and administration tasks handled by provider
  - Include high availability, upgrades, etc.
  - **Elastic scalability**: Clients pay exactly for the resources they consume; consumption can grow/shrink dynamically
    - No capital expenditures and fast provisioning
- Three different types exist at the moment
  - Simplified data management systems (e.g., Amazon SimpleDB)
  - Standard relational data management systems
  - Analysis services such as Amazon Elastic MapReduce

Outline

- Overview of three systems
  - Amazon Web Services with SimpleDB RDS, and Elastic MapReduce
  - Google App Engine with the Google App Engine Datastore
  - Microsoft Azure platform with Azure SQL

- Discussion
  - Technical challenges behind databases as a service
  - Broader impacts of databases as a service

Amazon Web Services

- Since 2006
- “Infrastructure web services platform in the cloud”

- Amazon Elastic Compute Cloud (Amazon EC2™)
- Amazon Simple Storage Service (Amazon S3™)
- Amazon SimpleDB™
- Amazon Elastic MapReduce™
- And more…

Amazon EC2

- Amazon Elastic Compute Cloud (Amazon EC2™)

- Rent compute power on demand ("server instances")
  - Select required capacity: small, large, or extra large instance
  - Share resources with other users (i.e., multi-tenant)
  - Variety of operating systems

- Includes: Amazon Elastic Block Store
  - Off-instance storage that persists independent from life of instance
  - Highly available and highly reliable

Amazon S3

- Amazon Simple Storage Service (Amazon S3™)
  - “Storage for the Internet”
  - “Web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.”

- Some key features
  - Write, read, and delete uniquely identified objects containing from 1 byte to 5 gigabytes of data each
  - Objects are stored in buckets, located in US or Europe
  - A bucket can be accessed from anywhere
  - Authentication
  - Reliability

Amazon SimpleDB

- "Web service providing the core database functions of data indexing and querying"

- **Partitioning**
  - Data partitioned into domains: queries run within domain
  - Domains seem to be unit of replication. Limit 10GB
  - Can use domains to manually create parallelism

- **Schema**
  - No fixed schema
  - Objects are defined with attribute-value pairs

Amazon SimpleDB (2/3)

- **Indexing**
  - Automatically indexes all attributes
- **Support for writing**
  - PUT and DELETE items in a domain
- **Support for querying**
  - GET by key
  - Selection + sort
  - A simple form of aggregation: count
  - Query is limited to 5s and 1MB output (but can continue)

```
select output_list
from domain_name
[where expression]
[sort_instructions]
[limit limit]
```

Amazon SimpleDB (3/3)

- **Availability and consistency**
  - "Fully indexed data is stored redundantly across multiple servers and data centers"
  - "Takes time for the update to propagate to all storage locations. The data will eventually be consistent, but an immediate read might not show the change"

- **Integration with other services**
  - "Developers can run their applications in Amazon EC2 and store their data objects in Amazon S3."
  - "Amazon SimpleDB can then be used to query the object metadata from within the application in Amazon EC2 and return pointers to the objects stored in Amazon S3."

Amazon RDS

- Amazon Relational Database Service (Amazon RDS^{TM})
  - Web service that facilitates set up, operations, and scaling of a relational database in the cloud
  - Full capabilities of a familiar MySQL database
- Some key features
  - Automated patches and backups for user-defined retention period
  - Elastic scalability of course
  - Different db instance sizes
- How do features and costs compare to SimpleDB?

Price Comparison

- Amazon RDS DB instance prices
  - From Small DB Instance $0.11/hour
  - To Quadruple Extra Large DB Instance $3.10/hour
  - $0.10 per GB-month of provisioned storage
  - $0.10 per 1 million I/O requests
- SimpleDB pricing
  - First 25 Amazon SimpleDB Machine Hours / month are free
  - $0.140/hour thereafter
  - First 1 GB of data transferred in/out per month is free
  - $0.100 per GB transferred in and $0.170 per GB out thereafter
  - First 1 GB stored per month is free
  - $0.250 per GB-month thereafter

Amazon Elastic MapReduce

- “Web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data”
- Hosted Hadoop framework on top of EC2 and S3
- Support for Hive and Pig
- User specifies
  - Data location in S3
  - Query
  - Number of machines
- System sets-up the cluster, runs query, and shuts down

Google App Engine

- “Run your web applications on Google's infrastructure”
- Limitation: applications must be written in Python or Java
- Key features (examples for Java)
  - A complete development stack that uses familiar technologies to build and host web applications
  - Includes: Java 6 JVM, a Java Servlets interface, and support for standard interfaces to the App Engine scalable datastore and services, such as JDO, JPA, JavaMail, and Jcache
  - JVM runs in a secured "sandbox" environment to isolate your application for service and security (some ops not allowed)

Google App Engine Datastore (1/3)

- “Distributed data storage service that features a query engine and transactions”

- **Partitioning**
  - Data partitioned into “entity groups”
  - Entities of the same group are stored together for efficient execution of transactions
- **Schema**
  - Each entity has a key and properties that can be either
    - Named values of one of several supported data types (includes list)
    - References to other entities
  - Flexible schema: different entities can have different properties

Google App Engine Datastore (2/3)

- **Indexing**
  - Applications define indexes: must have one index per query type

- **Support for writing**
  - PUT and DELETE entities (for Java, hidden behind JDO)

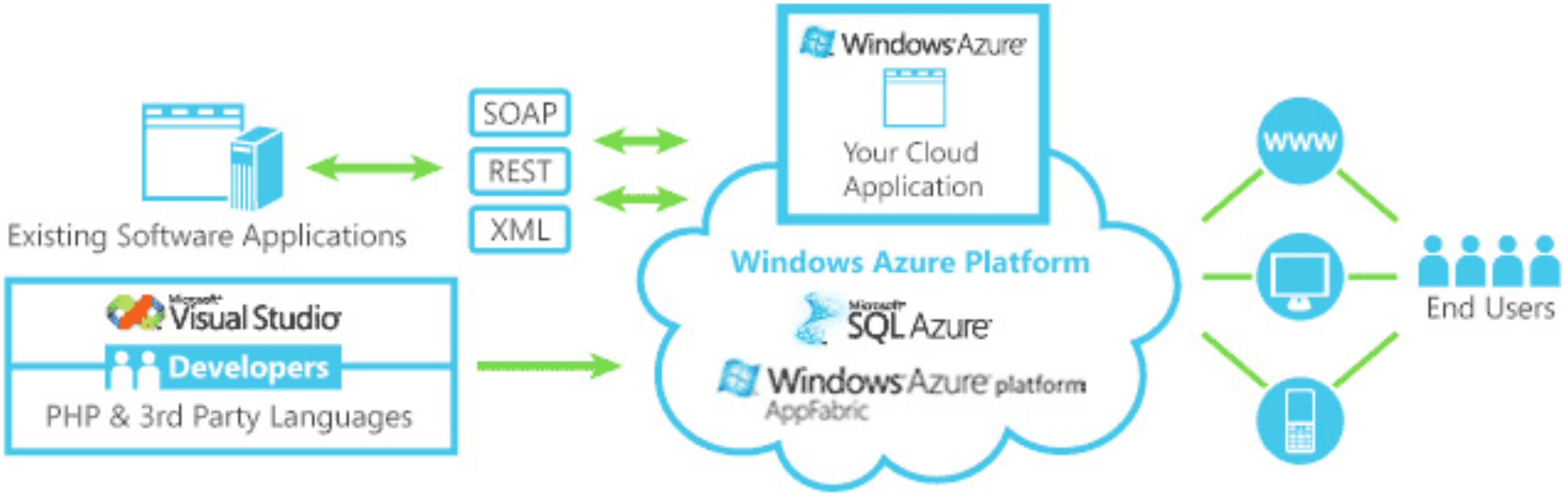
- **Support for querying**
  - Fetch an entity using its key
  - Execute a query: selection + sort
  - Language bindings: invoke methods or write SQL-like queries
  - Lazy query evaluation: query executes when user accesses results

Google App Engine Datastore (3/3)

- **Availability and consistency**
  - Every datastore write operation (put/delete) is atomic
  - Support transactions
    - All operations must operate on entities in the same entity group
  - Optimistic concurrency control

Microsoft Azure Platform

- "Internet-scale cloud computing and services platform"
- "Provides an operating system and a set of developer services that can be used individually or together"



Azure SQL

- "Cloud-based relational database service built on SQL Server® technologies"

- Key features
  - Highly available, scalable, multi-tenant database service
  - Includes authentication and authorization
  - No administration
  - Full-featured DBMS

- Key limitation
  - Only 10 GB at the moment

Outline

- Overview of three systems
  - Amazon Web Services with SimpleDB RDS, and Elastic MapReduce
  - Google App Engine with the Google App Engine Datastore
  - Microsoft Azure platform with Azure SQL
- Discussion
  - Technical challenges behind databases as a service
  - Broader impacts of databases as a service

Challenges of DBMS as a Service

- Scalability requirements
  - Large data volumes and large numbers of clients
  - Variable and heavy workloads
- High performance requirements: interactive web services
- Consistency and high availability guarantees
- Service Level Agreements
- Security

Broader Impacts

- Cost-effective solution for building web services
- Content providers focus only on their application logic
  - Service providers take care of administration
  - Service providers take care of operations
- Security/privacy concerns: all data stored in data centers