Cloud Computing

- In Wikipedia
  - Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them.
- Related keywords
  - Key-Value Store, MapReduce, Hadoop

In Industry

SaaS: Software as a Service
PaaS: Platform as a Service
IaaS: Infrastructure as a Service

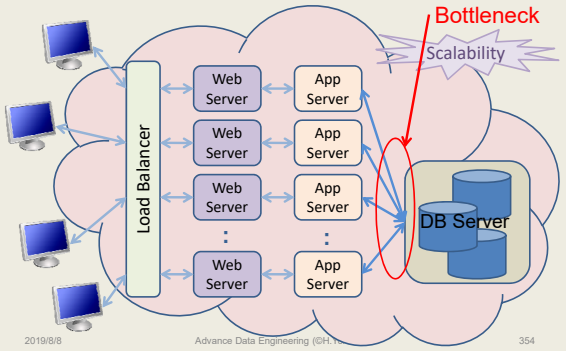
- Google: Gmail, AppEngine (Bigtable+Python)
- Yahoo!: PNUTS/Sherpa
- Amazon: S3, EC2
- Microsoft: Azure
- Sun: Sun Cloud
- IBM: XaaS
- …

[Source http://markusklems.wordpress.com/]

Requirements to a Cloud Center

- Scalability
  - "Major Issue" of the Cloud Computing
    - in CACM 2008 article by Brian Hayes
  - Scale-out performance with keeping dependability
    - Including Security & Privacy
- Flexible configuration transparent to clients
  - Virtualization
- Low energy consumption
- Hardware vs. Software
  - Of course hardware is important
  - But, software (especially database technology) plays more important role to satisfy them

Bottleneck of the N-tier Model



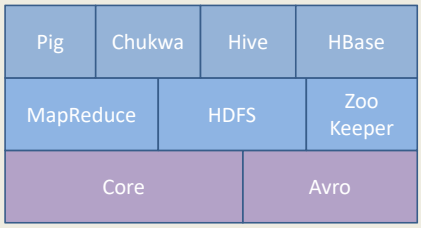
Available Software Related to KVS

- Apache Hadoop
  - Consist of a number of sub-projects
    - Include Hbase and Hive
- Apache Cassandra
  - Amazon's Dynamo + Google's Bigtable
    - Dynamo: Key-Value Store
    - Bigtable: ColumnFamily-base data model
- Voldemort
  - Distributed Key-Value Storage System
- …

Key-Value Store (KVS)

- Simple data model
  - Pairs of Key and Value (or Value List)
    - e.g. <car, 1>, <car, [1,1,1]>
      - car is key and 1 or [1,1,1] is value
- Main goal: Scalability
  - Does not support complex queries or aggregation
  - Neither supports strict consistency
    - Eventually consistency
- Applications: do not require ACID properties
  - Such as Google's search engine

Apache Hadoop Project



[Source: Hadoop: The Definitive Guide by Tome White O'REILLY]

Hadoop Subprojects

- **Hadoop Common**: The common utilities that support the other Hadoop subprojects.
- **Avro**: A data serialization system that provides dynamic integration with scripting languages.
- **Chukwa**: A data collection system for managing large distributed systems.
- **HBase**: A scalable, distributed database that supports structured data storage for large tables.
- **HDFS**: A distributed file system that provides high throughput access to application data.
- **Hive**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **MapReduce**: A software framework for distributed processing of large data sets on compute clusters.
- **Pig**: A high-level data-flow language and execution framework for parallel computation.
- **ZooKeeper**: A high-performance coordination service for distributed applications.

[Source http://hadoop.apache.org/]

MR Approach vs. Parallel DBMSs

- MapReduce and Parallel DBMSs: Friend or Foes?
  - Michael Stonebraker, Daniel Abadi, Devid J. Dewitt, Sam Madden, Erik Paulson, Andrew Pavlo, and Alexander Rasin
  - CACM, Vol. 53, No. 1, pp.64-71, Jan. 2010.

MapReduce⟶Cloud Approach

- Originally
  - A Comparison of Approaches to Large-Scale Data Analysis
  - Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Adabi, David J. Dewitt, Samuel Madden, Michael Stonebraker
  - SIGMOD'09, pp. 165-178, July 2009.

Differences

- MapReduce Approaches
  - Simple!
  - Suite for batch processes
    - Such as Extract-Transform-Load (ETL) system
- Parallel DBMS
  - Interactive query and update
  - Well optimized for structured information
- Conclusion
  - Complementary (not competitive)

Consistency

- Another important issue to compare DBMSs with new approaches
- **CAP Theorem**
  - It is trivial to achieve 2 out of the following 3:
    - **C**: Consistency (ACID trans. with serializability)
    - **A**: Availability (the service is always available)
    - **P**: Partition Tolerance (in network)
  - It is impossible to have all of them
    - Theoretically proved in [Brewer PODC2000, Gilbert & Lynch SIGACT News 2002]

CAP: DBMSs vs. New Approaches

- Tendency
  - Traditional DBMSs: sacrifice **A** (availability)
  - New approaches: sacrifice **C** (consistency)
- Why sacrifice consistency?
  - Nobody understands what sacrificing **P** means
  - Sacrificing **A** is unacceptable in the Web
  - **C** is not actually needed in many applications
    - Banks do not implement ACID (classic example wrong)
    - Airline reservation only transacts reads
    - Data is noisy and inconsistent anyway
      - making it, say, 1% worse does not matter

[Source: ICDE2010 Keynote by Donald Kossmann]

Comparison on consistency

	ACID	Pre-record ACD	Eventual	Other
ORACLE	✓	✓		Weaker serializable level
MySQL	✓	✓		Weaker serializable level
Vertica	✓	✓		
Cassandra		✓	✓	Support quorum read/write
HBase				Best effort
HDFS				Does not allow updates
UDB				Best effort-only

[Source: DASFAA2010 Keynote by Raghu Ramakrishnan]

Semantic Web

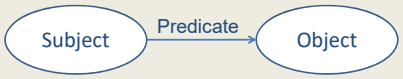
- The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries
  - The term was coined by Tim Berners-Lee for a web of data that can be processed by machines
  - Tim Berners-Lee: best known as the inventor of the World Wide Web

Important Components

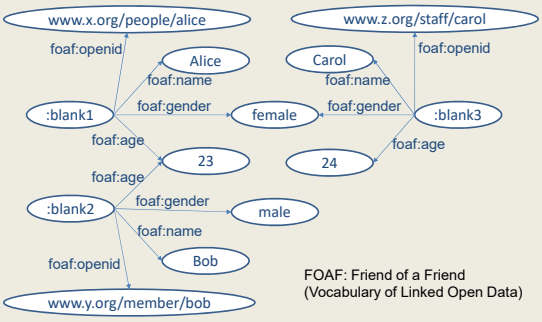
- XML: Extensible Markup Language
- RDF: Resource Description Framework
- RDFS: RDF Schema
  - provides a data-modelling vocabulary for RDF data
- SPARQL: SPARQL Protocol and RDF Query Language
- OWL: Web ontology language
- URI: Uniform Resource Identifier
- CRYPT: Cryptography technology

RDF Data Model

- Triple: <Subject, Predicate, Object>
  - Subject: to indicate a resource (an entity)
  - Predicate: to represent a property of the entity
  - Object: a value of the property in form of a resource or literal
- A set of triples builds a directed graph
  - Subject and object are nodes
  - Predicate is a labeled directed edge from the subject to the object



An RDF Graph Example



FOAF: Friend of a Friend
(Vocabulary of Linked Open Data)

RDF/XML

- RDF/XML representation

```
<rdf: RDF
  foaf = "http://xmlns.com/foaf/0.1/"
  <rdf Description rdf:about = "http://www.x.org/people/alice">
    <foaf:name>Alice</foaf:name>
    <foaf:gender>female</foaf:gender>
    <foaf:age>23</foaf:age>
  <rdf Description rdf:about = "http://www.y.org/member/bob">
    <foaf:name>Bob</foaf:name>
    <foaf:gender>male</foaf:gender>
    <foaf:age>23</foaf:age>
  ....
</rdf:RDF>
```

N3 Notations of RDF

- RDF data in N3 notation
  @prefix: foaf: <http://xmlns.com/foaf/0.1/>
  :blank1 foaf:openid <http://www.x.org/people/alice>
  :blank1 foaf:name “Alice”
  :blank1 foaf:gender “female”
  :blank1 foaf:age 23
  :blank2 foaf:opened <http://www.y.org/member/bob>
  :blank2 foaf:name “Bob”
  :blank2 foaf:gender “male”
  :blank2 foaf:age 23
  :blank3 foaf:openid <http://www.z.org/staff/carol>
  …
- There are other notations: N-Triple, Turtle, RDFa,…

SPARQL Query Language

- Recursive definition (W3C)
  – SPARQL Protocol and RDF Query Language
- Example
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  SELECT ?x
  WHERE {?y foaf:name ?x
  ?y foaf:age 23}
- Pattern match for <S, P, O>
- Self Join Operation

Graph Match by SPRQL

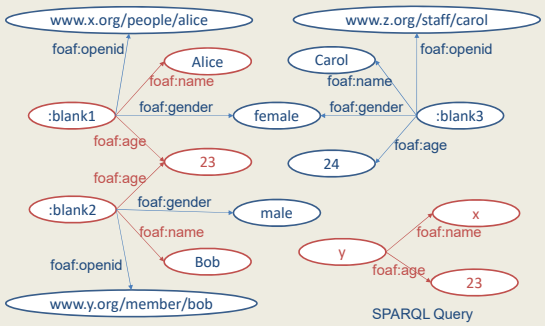
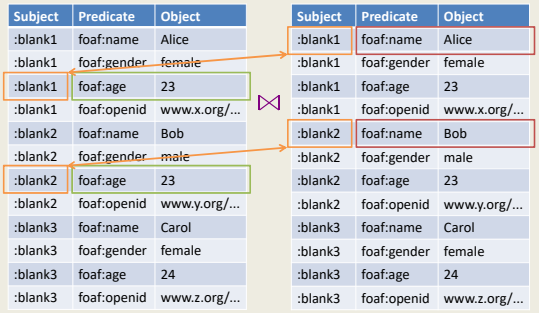


Image of Self Join



Subject	Predicate	Object
:blank1	foaf:name	Alice
:blank1	foaf:gender	female
:blank1	foaf:age	23
:blank1	foaf:openid	www.x.org/...
:blank2	foaf:name	Bob
:blank2	foaf:gender	male
:blank2	foaf:age	23
:blank2	foaf:openid	www.y.org/...
:blank3	foaf:name	Carol
:blank3	foaf:gender	female
:blank3	foaf:age	24
:blank3	foaf:openid	www.z.org/...

⋈

Subject	Predicate	Object
:blank1	foaf:name	Alice
:blank1	foaf:gender	female
:blank1	foaf:age	23
:blank1	foaf:openid	www.x.org/...
:blank2	foaf:name	Bob
:blank2	foaf:gender	male
:blank2	foaf:age	23
:blank2	foaf:openid	www.y.org/...
:blank3	foaf:name	Carol
:blank3	foaf:gender	female
:blank3	foaf:age	24
:blank3	foaf:openid	www.z.org/...

OWL (Web Ontology Language)

- A family of knowledge representation language as an extension of RDF vocabulary
  - OWL2 published by W3C OWL working group (2009)
    - OWL Lite, OWL DL (Description Logic), OWL Full
- RDF Schema (RDFS) is also ontology language
- Ex:

```
<owl:class rdf:ID="Mammal">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <owl:disjointWith rdf:resource="#Reptile"/>
</owl:Class>
```


Trends

- NoSQL (Not only SQL)
  - Note: Easy to misread, it does not deny SQL itself
- RDB: SQL
- OODB (Object-Oriented DB) : OQL (-> SQL)
  - ORDB (Object-Relational DB): PostgreSQL
- XML-DB: XQuery
- RDF: SPARQL (OWL/RDFS)
- KVS

Privacy and Security

- General concepts
  - Such as “privacy of an actor” and “home security”
  - Here, we focus on information processing
- Privacy
  - To seclude the information about an individual or group
    - Mainly by anonymization
- Security
  - To protect the stored and transferred information from leaks or corruptions
    - Mainly by cryptography

Anonymization

- To avoid to be identified a person (or a group) from given data
  - Identifier: passport number, student number, name, …
  - Linkable: tweet(time, GPS), blog(name, place) : name of tweeter
- In database
  - Anonymize by removing identifier (for OLAP)
  - Identify a person by combination of data
    - Example: Even though anonymize name, if you know the age of Alice and the database has only one tuple about a female whose age is 23…

name	age	gender	purchase
Alice	23	female	beef
Bob	23	male	beer
Carol	24	female	pork
David	22	male	milk

⇨

name	age	gender	purchase
*	23	female	beef
*	23	male	beer
*	24	female	pork
*	22	male	milk

k-anonymity [Latanya Sweeney 2002]

- k-anonymity: A release of data is said to have the *k*-anonymity property if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appear in the release.
- k-anonymization:

name	age	gender	purchase	
*	20< x <25	female	meat	k = 2
*	20< x <25	female	meat	
*	20< x <25	male	drinks	k = 2
*	20< x <25	male	drinks	

Cryptography (1/2)

- Long history
  - In ancient Egypt: secret hieroglyphics
  - In ancient Greece: the scytale of Sparta
  - In ancient Roma: Caesar cipher
    - Simply shifts the letters in the alphabet by a constant number of steps
    - "RETURN TO ROME" -> "UHWXUQ WR URPH"

ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFCHIJKLMNOPQRSTUVWXYZABC

In World War 2

German Enigma encryption machine

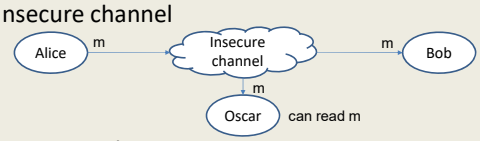
Alan Turning (The Imitation Game)

Cryptography (2/2)

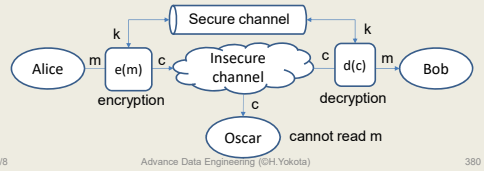
- Symmetric Algorithm
  - Traditional cryptography
- Asymmetric (or Public-Key) Algorithm
  - Introduced after 1976
- Cryptographic Protocols
  - Symmetric / Asymmetric algorithms are building blocks

Symmetric Cryptography

- Insecure channel

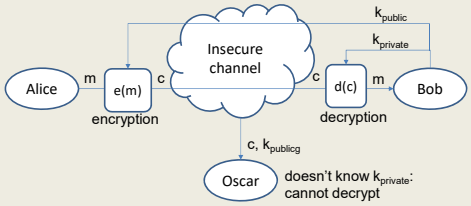


- Symmetric-key cryptosystem



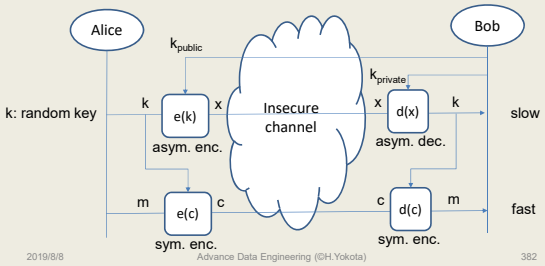
Asymmetric Cryptography

- Public-key cryptosystem (requires no secure channel)
  - Generate a pair of keys: public and private keys
  - Encrypt by public key, decrypted by private key
    - $c = e_{kpublic}(m)$, $m = d_{kprivate}(c)$



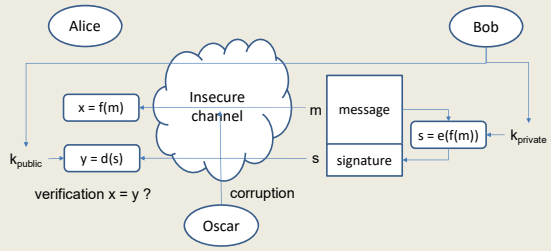
Key Transfer Protocol

- Symmetric key can be transferred using asymmetric cryptography



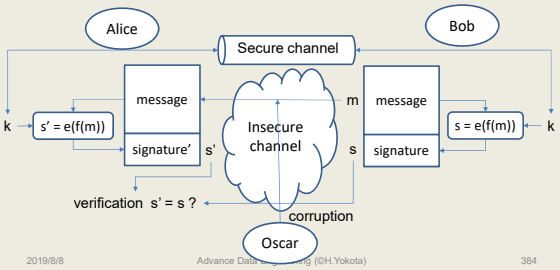
Digital Signature

- Only the person who creates a digital message should be capable of generating a valid signature.



Message Authentication Codes

- The difference between the message authentication codes (mac) and digital signature is that mac uses a symmetric key



Deterministic/Probablistic Encryption

- Deterministic Encryption
  $m_1 = m_2 \Rightarrow e(m_1) = e(m_2)$
  – Can be used for verification and search
  – Vulnerability
- Probablistic Encryption
  $m_1 = m_2 \Rightarrow e(m_1) \neq e(m_2)$
  – Cannot be used for verification or search

Cryptography Algorithms

- Symmetric
  – DES: Data Encryption Standard
  – AES: Advanced Encryption Standard
- Asymmetric
  – RSA: Ronald Rivest, Adi Shamir, Leonard Adleman
    - Based on large integer number factorization problem
  – Discrete Logarithm Problem based
    - Diffie-Hellman Key Exchange
  – Elliptic Curve Crhptosystem

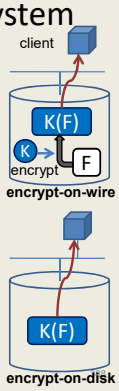
Security Levels

- If the best known attack requires 2^n steps
  - Security level of *n* bit

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

Encryption on a storage system

- Encryption schemes for the confidentiality on a network storage [Riedel et al., 2002]
  - Encrypt-on-wire scheme
    - Data is stored in clear, and encrypted when transmitted (e.g., SSL: Secure Socket Layer)
  - Encrypt-on-disk scheme
    - Data is stored in cipher, and transmitted without any encryption process
- Encrypt-on-disk scheme is more efficient than encrypt-on-wire scheme for the performance and confidentiality.
  - Storage server does not require as much encryption work with data transfer.
  - Encrypt-on-disk scheme protects data in storage while encrypt-on-wire scheme cannot.



Revocation on encrypt-on-disk (1/2)

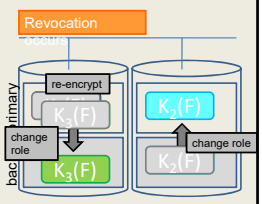
- With encrypt-on-disk, shared files must be re-encrypted when revocations occur.
  - There are possibilities of information leakage , if the revoked user holds the cryptographic key and intercepts the files.
- Re-encryption methods [Fu, 1999]
  - Active Revocation:
    - Files are re-encrypted immediately after the revocation.
    - It is enough secure
      - Revoked users are immediately unable to decrypt data
    - It has a problem of performance
      - Even authorized users cannot access them until re-encryptions are completed.

Revocation on encrypt-on-disk (2/2)

- Lazy Revocation:
  - The re-encryptions is delayed until the files are next updated
  - It is more efficient in respect of performance
    - Encryption involved in update process can be combined with the re-encryption required for revocations
    - The re-encryption work for multiple revocations are performed together if the file is not frequently updated
  - It is vulnerable
    - Data stored before update are still encrypted with the old key, which can be accessed by the revoked users.
- There is a trade-off problem between performance and security.

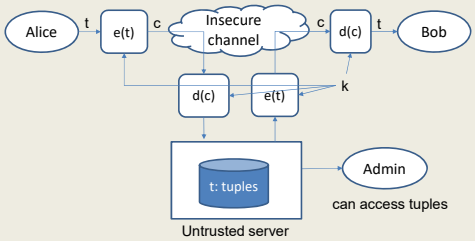
BA-Rev (Backup Assisted Revocation)

- We have proposed BA-Rev to attack the trade-off problem. [Takayama et al., 2007]
  - BA-Rev utilizes the primary-backup configuration.
- Outline
  1. Stores backup data with encrypted by key (K_2) different from that in primary (K_1)
  2. When a revocation occurs, their roles is changed
  3. Old primary data is re-encrypted by another key (K_3) and stored as backup
- Do not need wait for re-encryption
  - Authorized users can access the file immediately after the revocation.
  - Re-encryption processes are performed in background.



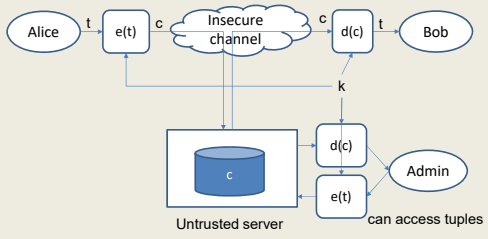
Untrusted Server

- Encrypted on wire approach
  - Malicious administrator



Untrusted Server

- Even encrypted on disk approach
  - If malicious administrator knows the key



Proxy Re-encryption

- Proxy re-encrypts cypher data