Προχωρημένη Κατανεμημένη Υπολογιστική

ΗΥ623

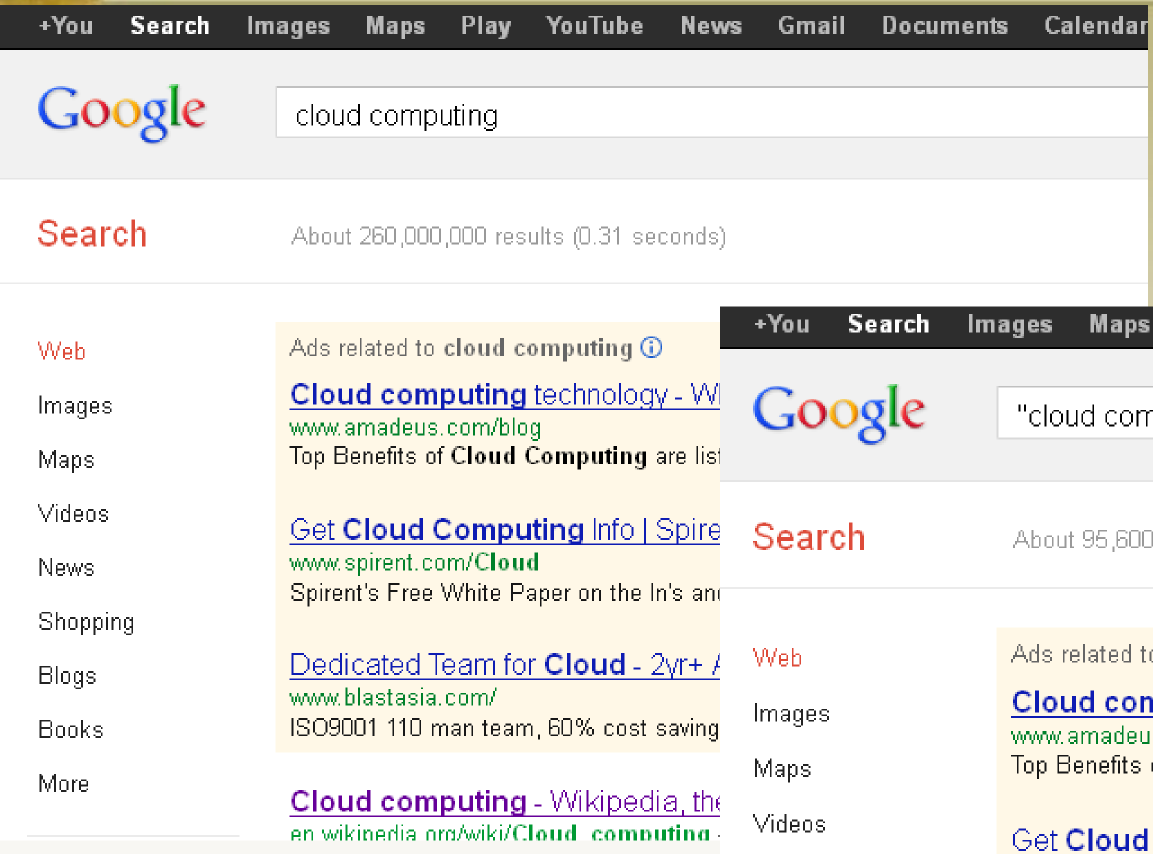
Διδάσκων –
Δημήτριος Κατσαρός

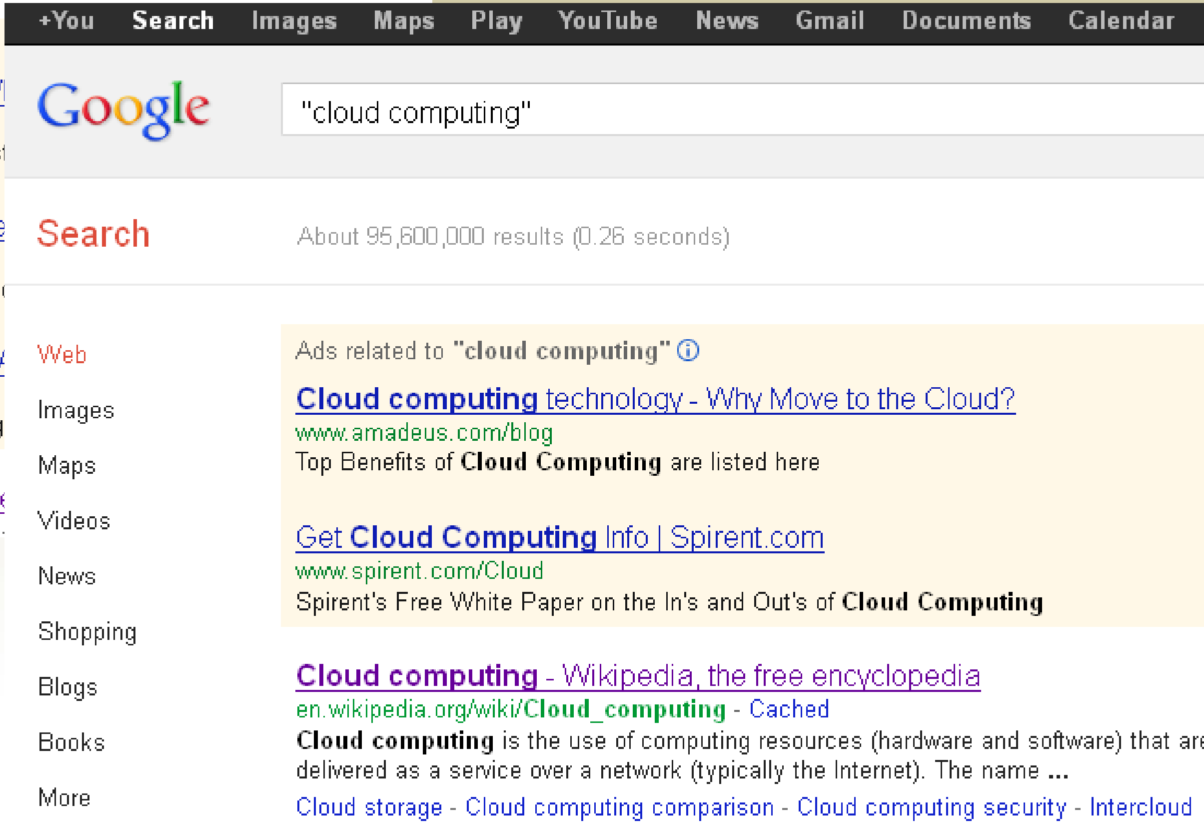
@ Τμ. Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ
Πανεπιστήμιο Θεσσαλίας

Cloud Computing

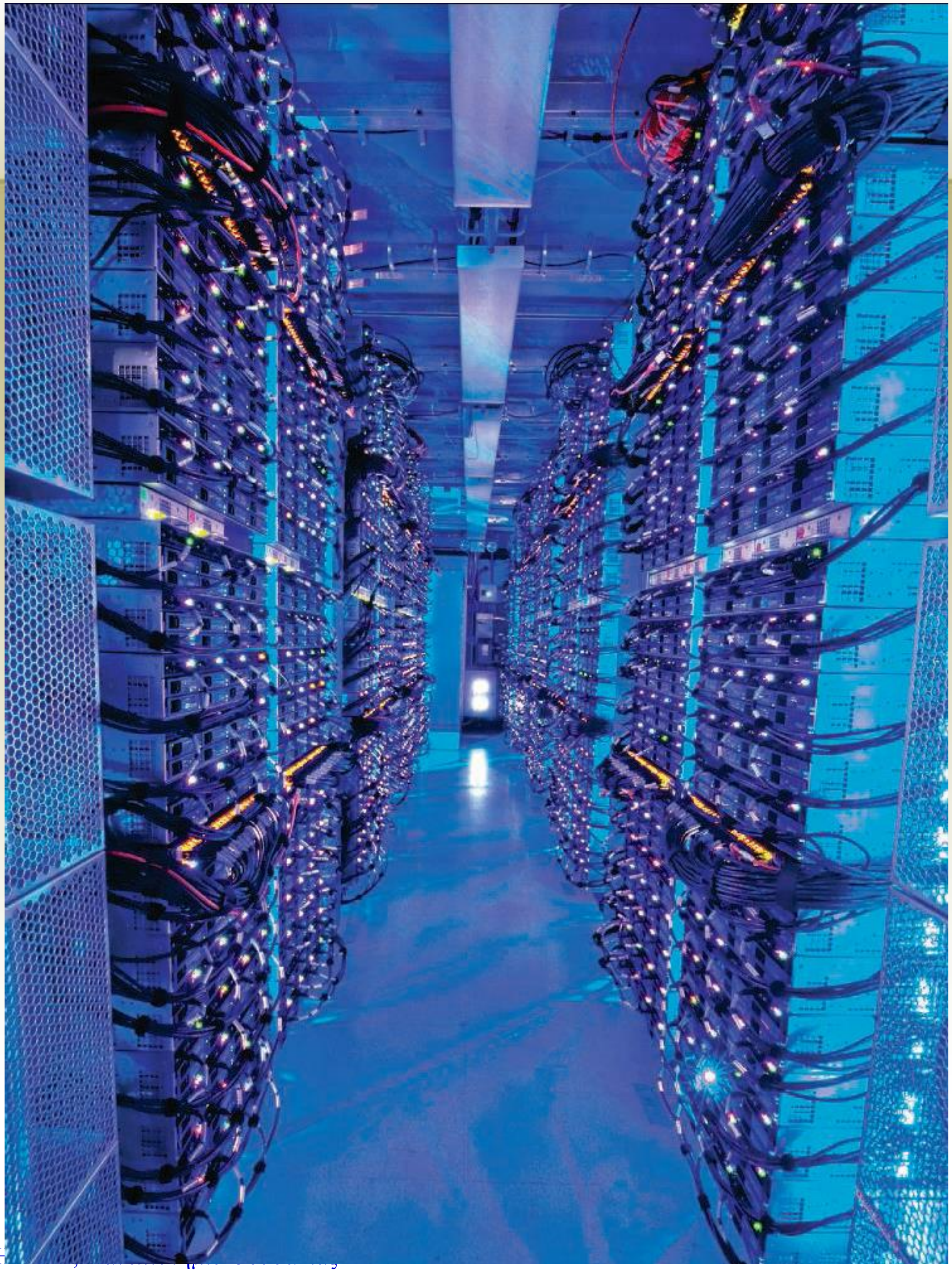
At a first glance

A buzz word …


+You
Search
Images
Maps
Play
YouTube
News
Gmail
Documents
Google
cloud computing
Search
About 260,000,000 results (0.31 seconds)
Web
Images
Maps
Videos
News
Shopping
Blogs
Books
More
Ads related to cloud computing
Cloud computing technology -
www.amadeus.com/blog
Top Benefits of Cloud Computing are
Get Cloud Computing Info |
www.spirent.com/Cloud
Spirent's Free White Paper on the In's
Dedicated Team for Cloud - 2yr+
www.blastasia.com/
ISO9001 110 man team, 60% cost
Cloud computing - Wikipedia,


+You
Search
Images
Maps
Play
YouTube
News
Gmail
Documents
Calendar
Google
"cloud computing"
Search
About 95,600,000 results (0.26 seconds)
Web
Images
Maps
Videos
News
Shopping
Blogs
Books
More
Ads related to "cloud computing"
Cloud computing technology - Why Move to the Cloud?
www.amadeus.com/blog
Top Benefits of Cloud Computing are listed here
Get Cloud Computing Info | Spirent.com
www.spirent.com/Cloud
Spirent's Free White Paper on the In's and Out's of Cloud Computing
Cloud computing - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Cloud_computing - Cached
Cloud computing is the use of computing resources (hardware and software) that
delivered as a service over a network (typically the Internet). The name ...
Cloud storage - Cloud computing comparison - Cloud computing security - Intercloud

The scale …



The investment …

Defining cloud computing

- Cloud computing refers to both
  - The applications delivered *as services* over the Internet, and
  - The hardware and systems software in the data centers that provide these services

- The term "grid computing" suggests
  - protocols to offer shared computation and storage over long distances,
  - but those protocols *did not lead to a software environment that grew* beyond its community

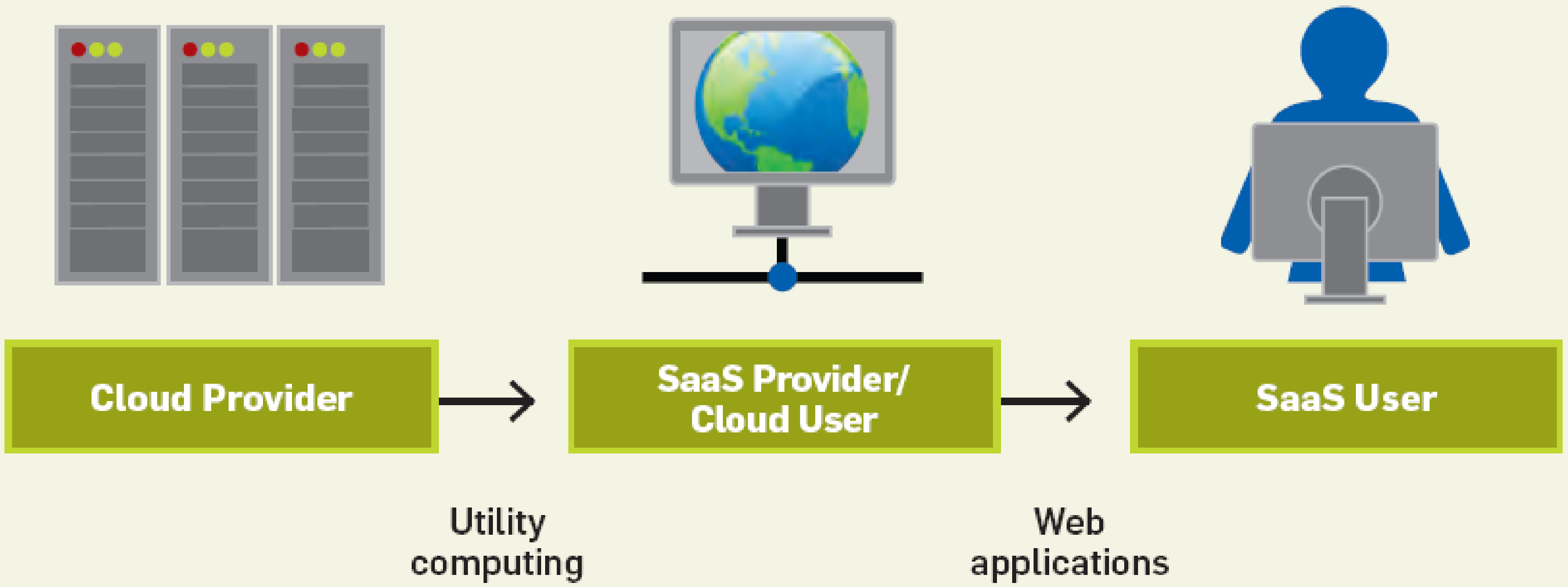
Definition by NIST

- "…a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

What is a cloud?

- The data center hardware and software is what we will call a *cloud*
- *Public cloud*: When a cloud is made available in a *pay-as-you-go* manner to the general public
  - the service being sold is *utility computing*
- *Private cloud*: refers to internal data centers of a business or other organization, not made available to the general public, when they are large enough to benefit from the advantages of cloud computing

Provider-user relationships



New aspects in cloud computing

- From a hardware provisioning and pricing point of view, three aspects are new in cloud computing:
  - The appearance of infinite computing resources available on demand, quickly enough to follow load surges, thereby eliminating the need for cloud computing users to plan far ahead for provisioning
  - The elimination of an up-front commitment by cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs
  - The ability to pay for use of computing resources on a short-term basis as needed (for example, processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful

Is everything “cloud computing”?

- Oracle's CEO Larry Ellison said once:

  "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."

Enablers of cloud computing

- Key necessary enabler
  - Construction and operation of extremely large-scale (**hundreds of thousands**), commodity-computer data centers at low-cost locations
- Uncovered the factors of 5 to 7 decrease in cost
  - electricity
  - network bandwidth
  - operations
  - software
  - hardware
  - Statistical multiplexing

Example of what is not CC

- Consider a public-facing Internet service hosted on an ISP who can allocate more machines to the service given four hours notice
  - Since load surges on the public Internet can happen much more quickly than that (Animoto saw its load double every 12 hours for nearly three days), <u>this is not cloud computing</u>

Public clouds vs. private data centers

Advantage	Public Cloud	Conventional Data Center
Appearance of infinite computing resources on demand	Yes	No
Elimination of an up-front commitment by Cloud users	Yes	No
Ability to pay for use of computing resources on a short-term basis as needed	Yes	No
Economies of scale due to very large data centers	Yes	Usually not
Higher utilization by multiplexing of workloads from different organizations	Yes	Depends on company size
Simplify operation and increase utilization via resource virtualization	Yes	No

Classes of cloud computing 1/7

- Infrastructure as a Service (IaaS)

  Provision resources such as servers (often in the form of virtual machines), network bandwidth, storage, and related tools necessary to build an application environment from scratch (e.g. Amazon EC2)

Classes of cloud computing 2/7

- Platform as a Service (PaaS)

  Provides a higher-level environment where developers can write customized applications (e.g., Microsoft Azure, Google AppEngine). The maintenance, load-balancing and scale-out of the platform are done by the service provider and the developer can concentrate on the main functionalities of his application

Classes of cloud computing 3/7

- Software as a Service (SaaS)

  Refers to special-purpose software made available through the Internet (e.g., Sales-Force). Therefore, it does not require each end-user to manually download, install, configure, run or use the software applications on their own computing environments

Classes of cloud computing 4/7

Application Service (SaaS)	MS Live/ExchangeLabs, IBM, Google Docs; Salesforce.com Quicken Online, Zoho, Cisco
Application Platform (Paas)	Google App Engine, Mosso, Force.com, Engine Yard, Facebook, Heroku, AWS
Server Platform (IaaS)	3Tera, EC2, SliceHost, GoGrid, RightScale, Linode
Storage Platform IaaS	Amazon S3, Dell, Apple, ...

Classes of cloud computing 5/7

- Amazon EC2 is at one end of the spectrum
  - an EC2 instance looks much like physical hardware
  - users can control nearly the entire software stack, from the kernel upward
  - This low level makes it inherently difficult for Amazon to offer automatic scalability and failover
    - because the semantics associated with replication and other state management issues are highly application-dependent

Classes of cloud computing 6/7

- Google AppEngine is at the other extreme of the spectrum
  - it is an application domain-specific platform
  - is targeted exclusively at traditional Web applications
  - enforces an application structure of clean separation between a stateless computation tier and a stateful storage tier
  - AppEngine's impressive automatic scaling and high-availability mechanisms, and the proprietary MegaStore data storage available to AppEngine applications, all rely on these constraints

Classes of cloud computing 7/7

- Applications for Microsoft's Azure are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment.
- The framework is significantly more flexible than AppEngine's, but still constrains the user's choice of storage model and application structure.
- Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2

Cloud computing economics

- Three compelling use cases that favor cloud computing
- A first case is when demand for a service varies with time
  - For example, provisioning a data center for the peak load it must sustain a few days per month leads to underutilization at other times.
  - Instead, cloud computing lets an organization pay by the hour for computing resources, potentially leading to cost savings even if the hourly rate to rent a machine from a cloud provider is higher than the rate to own one

Cloud computing economics

- A second case is when demand is unknown in advance
  - For example, a Web startup will need to support a spike in demand when it becomes popular, followed potentially by a reduction once some visitors turn away

Cloud computing economics

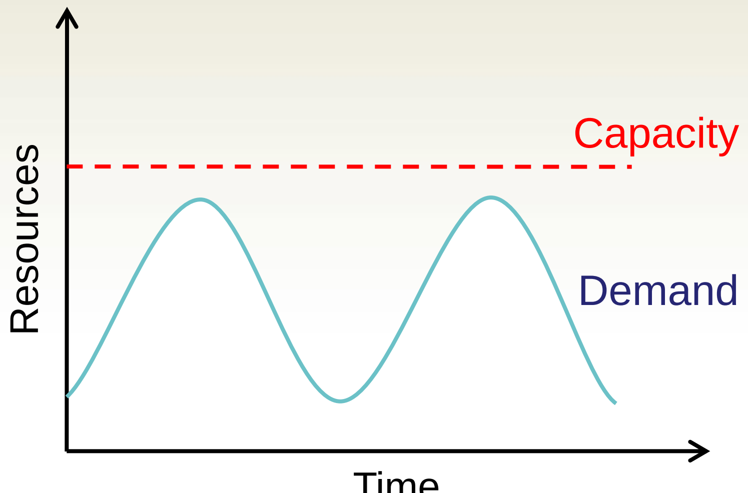
- A third case is for organizations that perform batch analytics
  - They can use the "cost associativity" of cloud computing to finish computations faster
  - using 1,000 EC2 machines for one hour costs the same as using one machine for 1,000 hours

The concept of *elasticity*

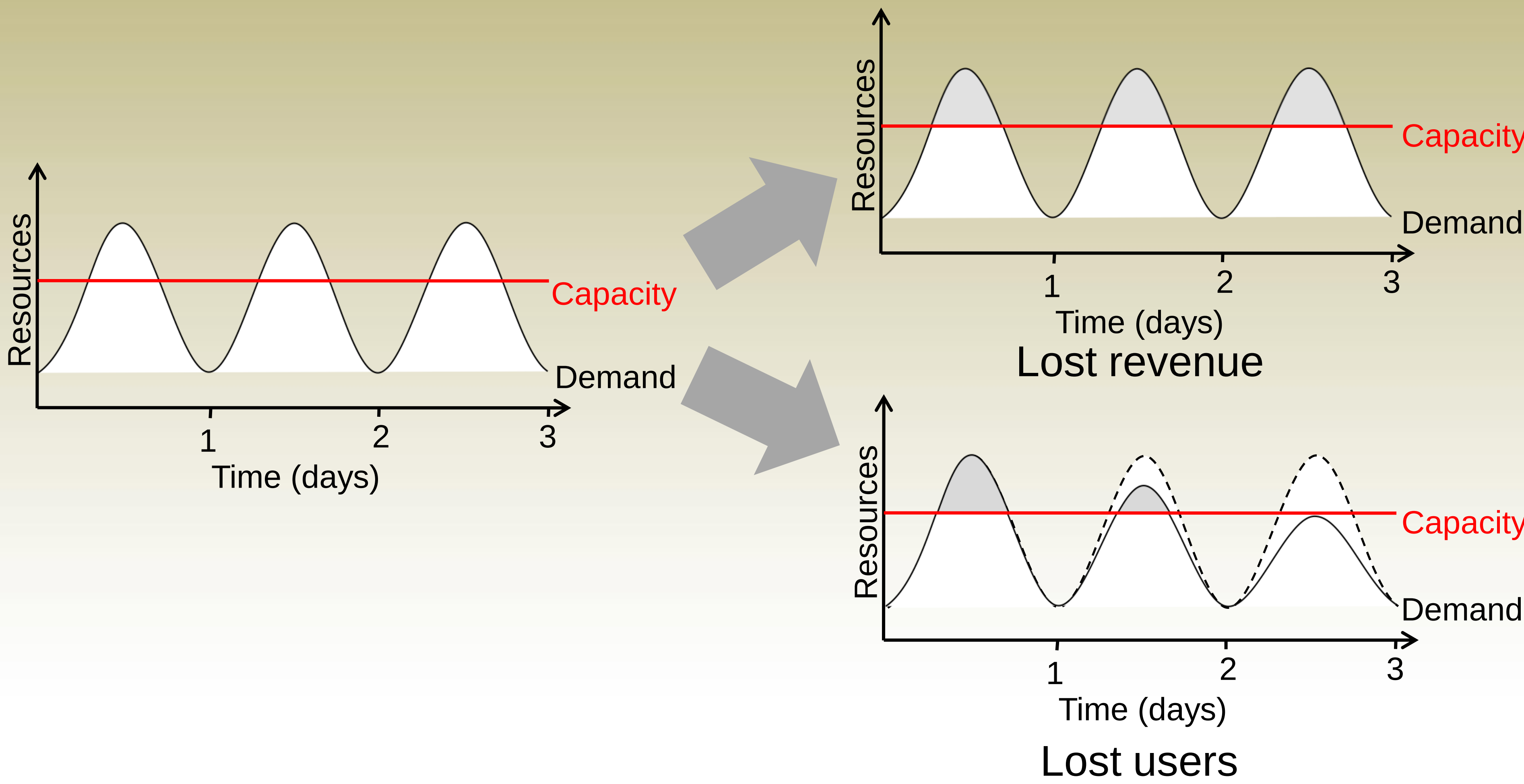
- Cloud computing's ability to
  - add or remove resources at a fine grain (e.g., one server at a time with EC2) and
  - with a lead time of minutes rather than weeks
- allows matching resources to workload much more closely

The cost of over-provisioning

- A service has a predictable demand where the peak requires 500 servers at noon and only 100 servers at midnight
- If average utilization over a whole day is 300 servers, the actual cost per day is $300 \times 24 = 7{,}200$ server hours
- must provision to the peak of 500 servers, we pay for $500 \times 24 = 12{,}000$ server-hours a factor of 1.7 more
- Therefore, as long as the pay-as-you-go cost per server-hour over three years (typical amortization time) is less than 1.7 times the cost of buying the server, utility computing is cheaper



The cost of under-provisioning



Adoption Challenges

Challenge	Opportunity
Availability	Multiple providers & DCs
Data lock-in	Standardization
Data Confidentiality and Auditability	Encryption, VLANs, Firewalls; Geographical Data Storage

Growth Challenges

Challenge	Opportunity
Data transfer bottlenecks	FedEx-ing disks, Data Backup/Archival
Performance unpredictability	Improved VM support, flash memory, scheduling VMs
Scalable storage	Invent scalable store
Bugs in large distributed systems	Invent Debugger that relies on Distributed VMs
Scaling quickly	Invent Auto-Scaler that relies on ML; Snapshots

Policy & Business Challenges

Challenge	Opportunity
Reputation Fate Sharing	Offer reputation-guarding services like those for email
Software Licensing	Pay-for-use licenses; Bulk use sales

Business Continuity & Service Availability

Service and Outage	Duration	Date
S3 outage: authentication service overload leading to unavailability[17]	2 hours	2/15/08
S3 outage: Single bit error leading to gossip protocol blowup[18]	6–8 hours	7/20/08
AppEngine partial outage: programming error[19]	5 hours	6/17/08
Gmail: site unavailable due to outage in contacts system[11]	1.5 hours	8/11/08

- How to achieve “no single point of failure”
- Even if the company has multiple data centers in different geographic regions using different network providers, it may have common software infrastructure and accounting systems, or the company may even go out of business

Data Lock-In

- Software stacks have improved interoperability among platforms
- Storage APIs for cloud computing are still essentially proprietary, or at least have not been the subject of active standardization
- An online storage service called The Linkup shut down on Aug. 8, 2008 after losing access as much as 45% of customer data
- The Linkup, in turn, had relied on the online storage service Nirvanix to store customer data, which led to finger pointing between the two organizations as to why customer data was lost. Meanwhile, The Linkup's 20,000 users were told the service was no longer available and were urged to try out another storage site

Data Confidentiality/Auditability

- Cloud users face security threats both from outside and inside the cloud
- Many of the security issues involved in protecting clouds from outside threats are similar to those already facing large data centers
- In the cloud, however, this responsibility is divided among potentially many parties, including the cloud user, the cloud vendor, and any third-party vendors that users rely on for security-sensitive software or configurations

Data Confidentiality/Auditability

- The cloud user is responsible for application-level security
- The cloud provider is responsible for physical security, and likely for enforcing external firewall policies
- Security for intermediate layers of the software stack is shared between the user and the operator; the lower the level of abstraction exposed to the user, the more responsibility goes with it
- Amazon EC2 users have more technical responsibility for their security than do Azure users, who in turn have more responsibilities than AppEngine customers

Data Confidentiality/Auditability

- Cloud computing poses the new problem of internal-facing security
- Cloud providers must guard against theft or denial-of-service attacks by users
- Users need to be protected from one another
- The primary security mechanism in today's clouds is virtualization
- It is a powerful defense, and protects against most attempts by users to attack one another or the underlying cloud infrastructure
- However, not all resources are virtualized and not all virtualization environments are bug-free

Data Transfer Bottlenecks

- To overcome the high cost of Internet transfers is to ship disks
- To quantify the argument, assume that we want to ship 10TB from U.C. Berkeley to Amazon in Seattle, WA
- Garfinkel9 measured bandwidth to S3 from three sites and found an average write bandwidth of 5Mbits/sec to 18Mbits/sec. Suppose we get 20Mbits/sec over a WAN link. It would take 10 * 1012 Bytes / (20×106 bits/second) = (8×1013)/(2×107) seconds = 4,000,000 seconds, which is more than 45 days
- If we instead sent 10 1TB disks via overnight shipping, it would take less than a day to transfer 10TB, yielding an effective bandwidth of about 1,500Mbit/sec

Performance Unpredictability

- Multiple virtual machines (VMs) can share CPUs and main memory surprisingly well in cloud computing, but that network and disk I/O sharing is more problematic
- Measured 75 EC2 instances running a memory benchmark
- The mean bandwidth is 1,355Mbytes/ sec., with a standard deviation across instances of just 52MBytes/sec, less than or about 4% of the mean
- Measured the average disk bandwidth for 75 EC2 instances each writing 1GB files to local disk
- The mean disk write bandwidth is nearly 55Mbytes per second with a standard deviation across instances of a little over 9MBytes/sec, or about 16% of the mean
- This demonstrates the problem of I/O interference between virtual machines

Scalable Storage

- Properties whose combination gives cloud computing its appeal:
  - short-term usage (which implies scaling down as well as up when demand drops)
  - no upfront cost
  - infinite capacity on demand
- While it's straightforward what this means when applied to computation, it's less clear how to apply it to persistent storage

Bugs in Large-Scale Distributed Systems

- A common occurrence is that such bugs cannot be reproduced in smaller configurations

Scaling Quickly

- Google AppEngine automatically scales in response to load increases and decreases, and users are charged by the cycles used
- AWS charges by the hour for the number of instances you occupy, even if your machine is idle
- The opportunity is then to automatically scale quickly up and down in response to load in order to save money, but without violating service-level agreements

Reputation Fate Sharing

- One customer's bad behavior can affect the reputation of others using the same cloud
- For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit which applications can be effectively hosted

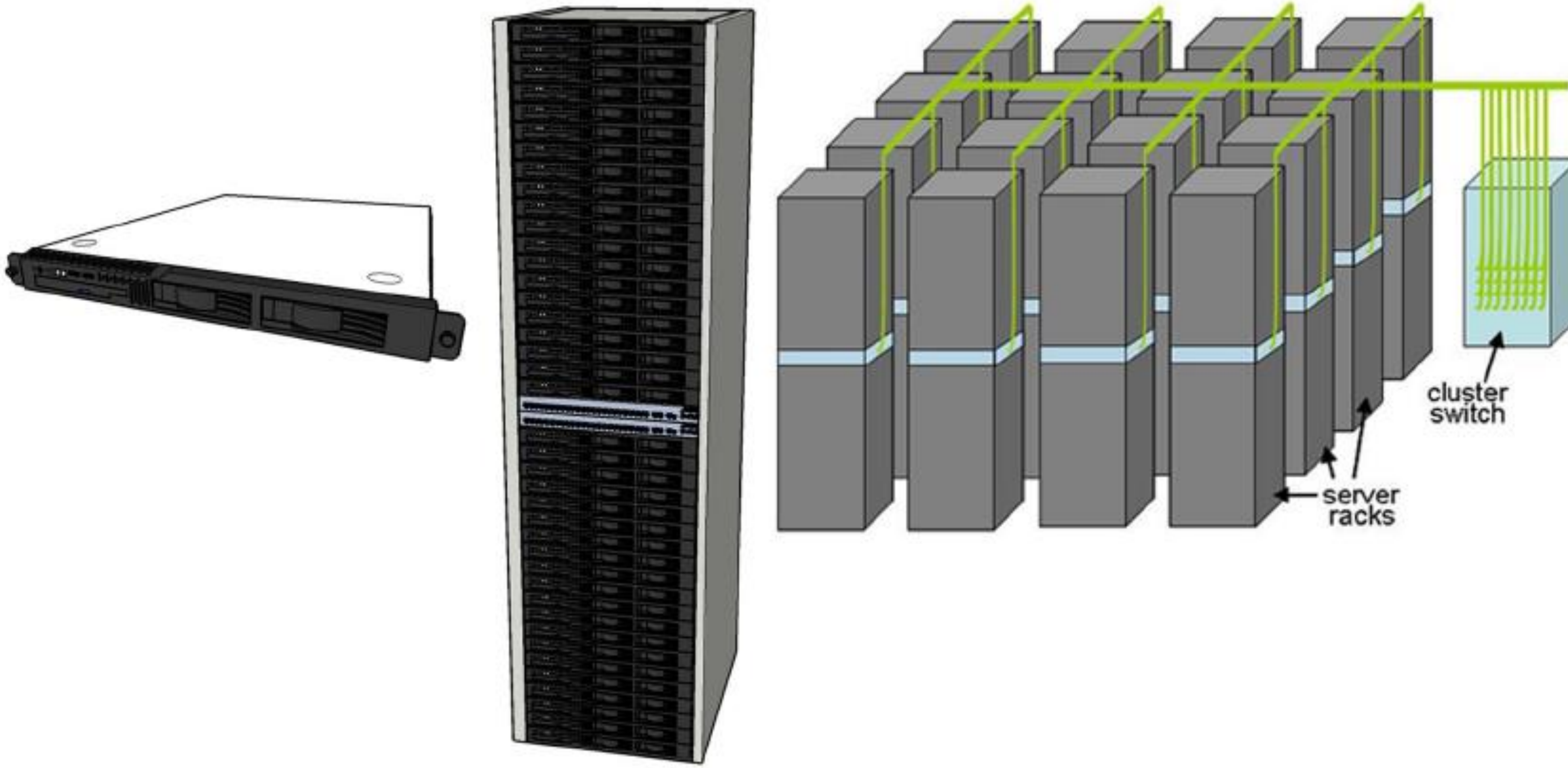
Software Licensing

- Current software licenses commonly restrict the computers on which the software can run. Users pay for the software and then pay an annual maintenance fee
- Hence, many cloud computing providers originally relied on open source software in part because the licensing model for commercial software is not a good match
- Software companies must change their licensing structure to better fit cloud computing. For example, Microsoft and Amazon now offer pay-as-you-go software licensing for Windows Server and Windows SQL Server on EC2

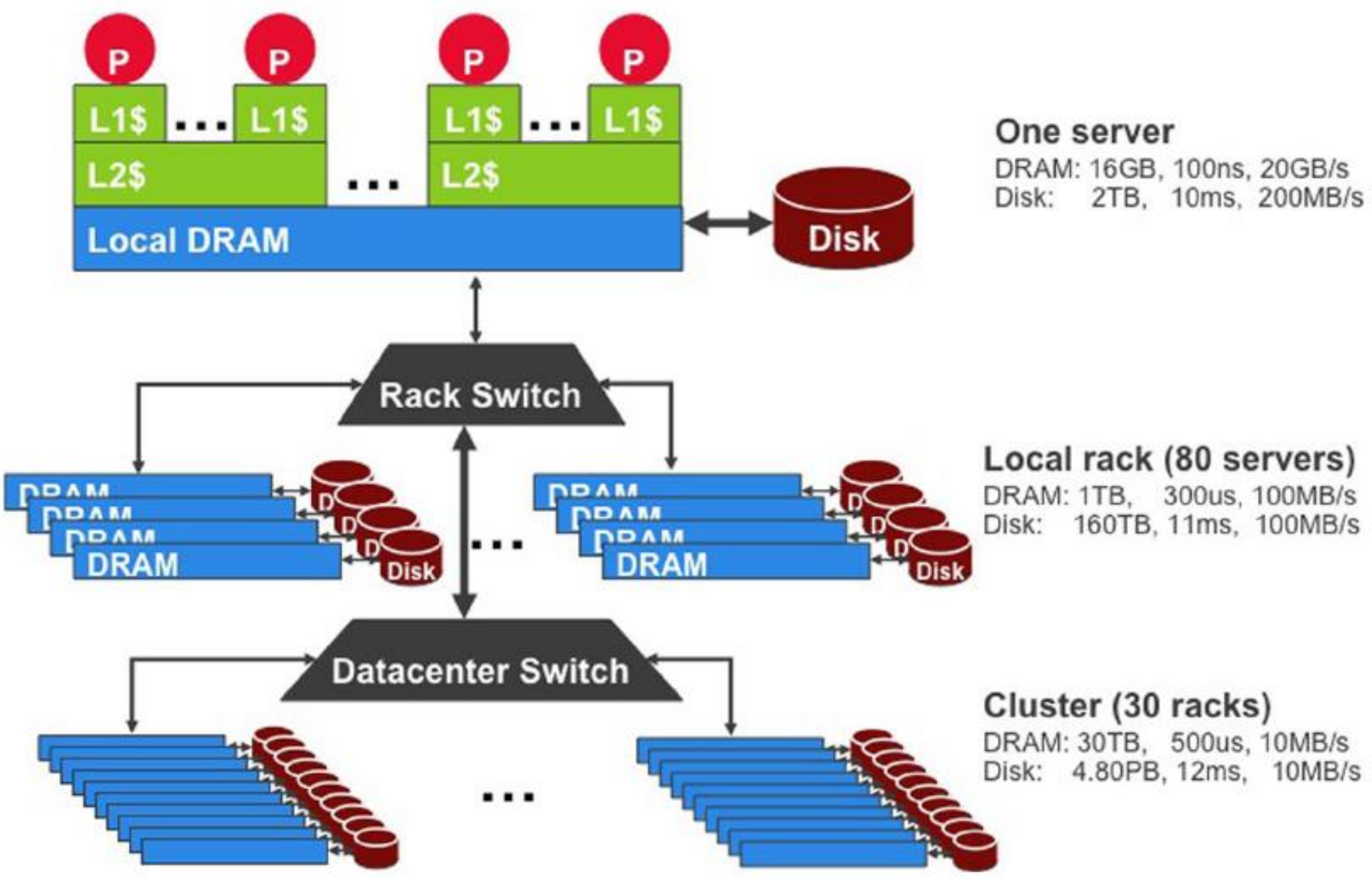
Large-scale Data Centers
OR
Warehouse-Scale Computers

Architectural overview

Typical elements in warehouse-scale systems



Storage hierarchy of a WSC



Software layers in a typical WSC

Platform-level software

The common firmware, kernel, operating system distribution, and libraries expected to be present in all individual servers to abstract the hardware of a single machine and provide basic server-level services

Software layers in a typical WSC

Cluster-level infrastructure

- The collection of distributed systems software that manages resources and provides services at the cluster level; we consider these services as an operating system for a datacenter
- Examples are distributed file systems, schedulers, remote procedure call layers, as well as programming models that simplify the usage of resources at the scale of datacenters, such as MapReduce, Dryad, Hadoop, Sawzall, BigTable, Dynamo, and Chubby

Software layers in a typical WSC

Application-level software

- Software that implements a specific service
- Online services
  - Google search, Gmail, and Google Maps
- Offline computations
  - Offline computations are typically used in large-scale data analysis or as part of the pipeline that generates the data used in online services
    - building an index of the Web
    - processing satellite images to create map tiles for the online service

Performance & availability toolbox

Concept	Performance	Availability
Replication	Yes	Yes
Sharding (partitioning)	Yes	Yes
Load-balancing	Yes	
Health checking & watchlog timers		Yes
Integrity checks		Yes
Application-specific compression	Yes	
Eventual consistency	Yes	Yes

Cluster-level infrastructure software

Resource management

- Controls the mapping of user tasks to hardware resources, enforces priorities and quotas, and provides basic task management services
  - In its simplest form, it can be an interface to manually (and statically) allocate groups of machines to a given user or job
- Users of such systems should be able to specify their job requirements at a relatively high level (e.g., how much CPU performance, memory capacity, networking bandwidth, etc.) and have the scheduler translate those into an appropriate allocation of resources
- It is increasingly important that cluster schedulers also consider power limitations and energy usage optimization

Cluster-level infrastructure software

Hardware abstraction & other basic services

- Every large-scale distributed application needs a small set of basic functionalities
- Examples are reliable distributed storage, message passing, and cluster-level synchronization
- It is wise to avoid re-implementing such tricky code for each application and instead create modules or services that can be reused
- GFS, Dynamo, and Chubby are examples of reliable storage and lock services for large clusters developed at Google and Amazon

Cluster-level infrastructure software

Deployment and maintenance

- Software image distribution and configuration management, monitoring service performance and quality, require a significant amount of infrastructure
- The Autopilot system from Microsoft offers an example design for some of this functionality for Windows Live datacenters
- Google's System Health Infrastructure is an example of the software infrastructure needed for efficient health management
- The X-Trace system is an example of monitoring infrastructure aimed at performance debugging of large distributed systems

Cluster-level infrastructure software

Programming frameworks

- From a programmer's standpoint, hardware clusters have a deep and complex memory/storage hierarchy, heterogeneous components, failure-prone components, and scarcity of some resources (such as DRAM and datacenter-level networking bandwidth)
- MapReduce, BigTable and Dynamo are good examples of pieces of infrastructure software that greatly improve programmer productivity by automatically handling data partitioning, distribution, and fault tolerance within their respective domains