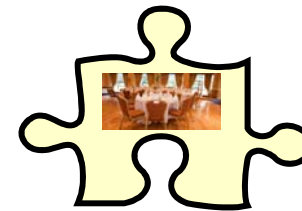
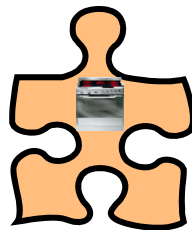
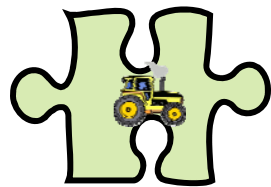


X as a Service

(IaaS, PaaS & SaaS)



Knowledge objectives

1. IaaS

1. Give a definition of Cloud Computing
2. Justify the usage of IaaS from an economical point of view
3. Enumerate some advantages of IaaS compared to on-premise infrastructure
4. Explain the difficulties of providing IaaS
5. Compare scalability and elasticity

2. PaaS

1. Enumerate several service provider platforms
2. Enumerate several benefits of software development in the cloud
3. Enumerate eight features of cloud software
4. Distinguish between cloud software (i.e., DBMS) and software in the cloud
5. Explain the importance of elasticity and how it can be achieved

3. SaaS

1. Explain three different cloud stacks
2. Explain the prevalence of different kinds of software as a service
3. Define multi-tenancy
4. Enumerate main multi-tenancy challenges



Understanding Objectives

1. Use the Universal Scalability Law to decide the number of resources needed
2. Compare benefits of on-premises against as-a-service software
3. Analyze the three different implementations of multi-tenancy software

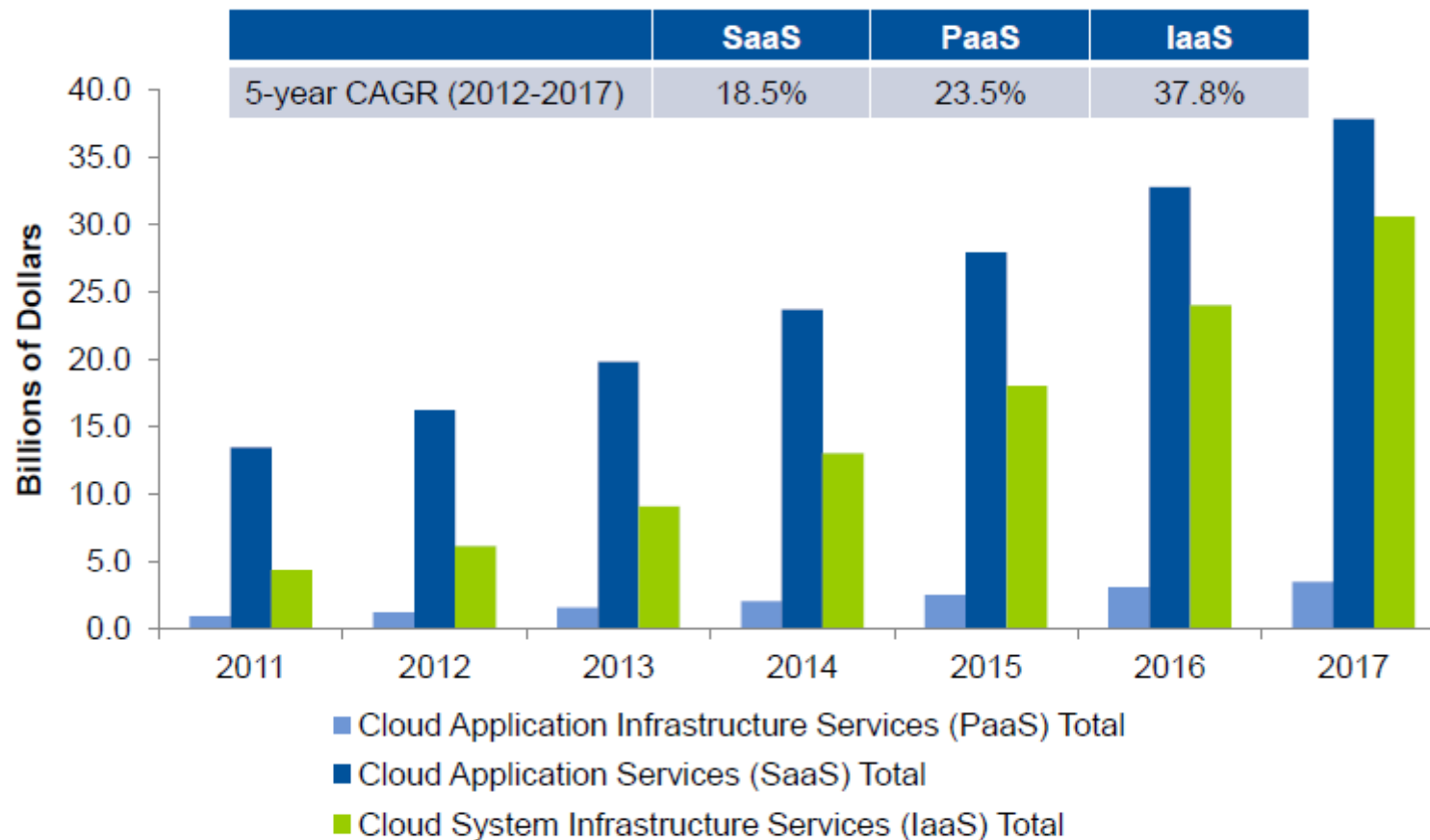


Application Objectives

1. Discuss whether some characteristics of IaaS are obstacles or opportunities to BI
2. Given some characteristics, select an IaaS/PaaS provider in front of others



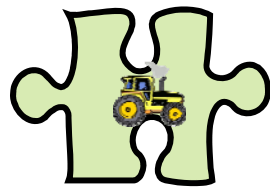
Gartner's forecast for XaaS



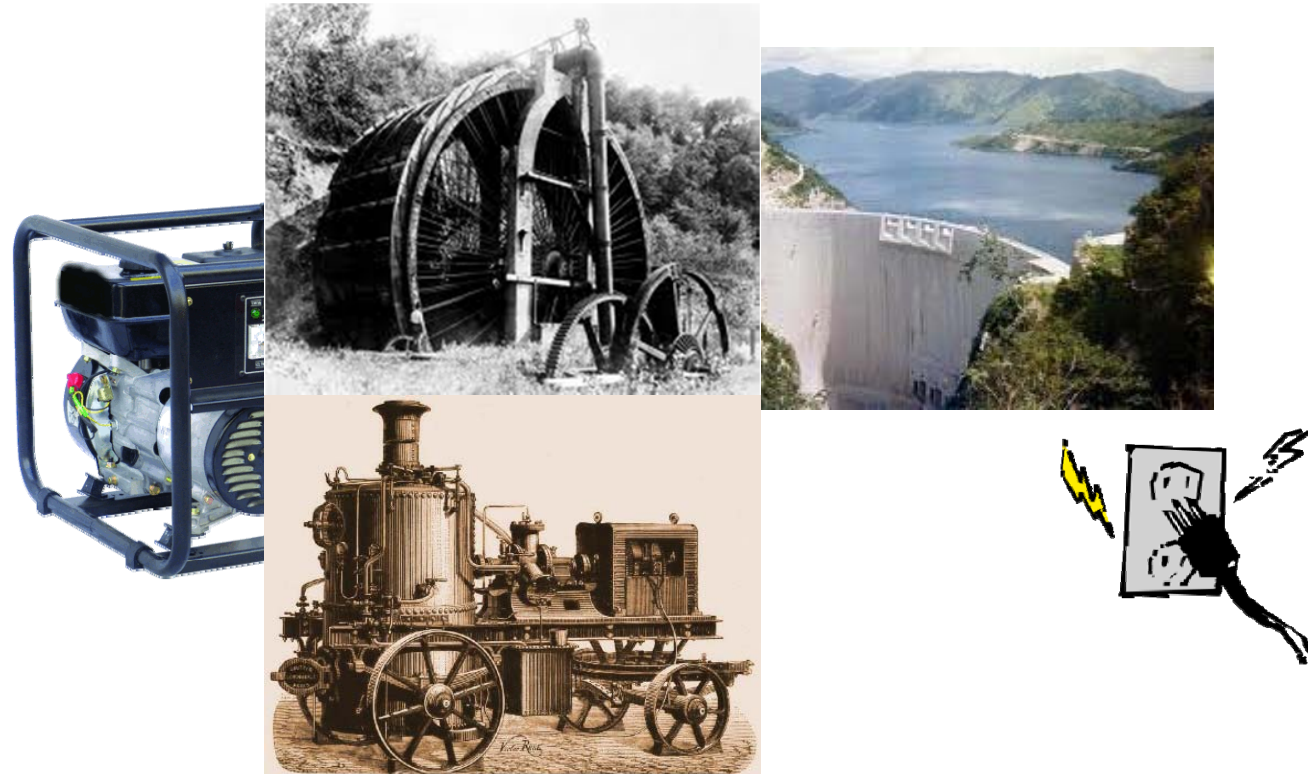
Sources: "IT Spending Forecast, 1Q13 Update", "Forecast: Public Cloud Services, Worldwide, 2011-2017, 1Q13 Update" (G00248727)



IaaS



Electricity as a utility

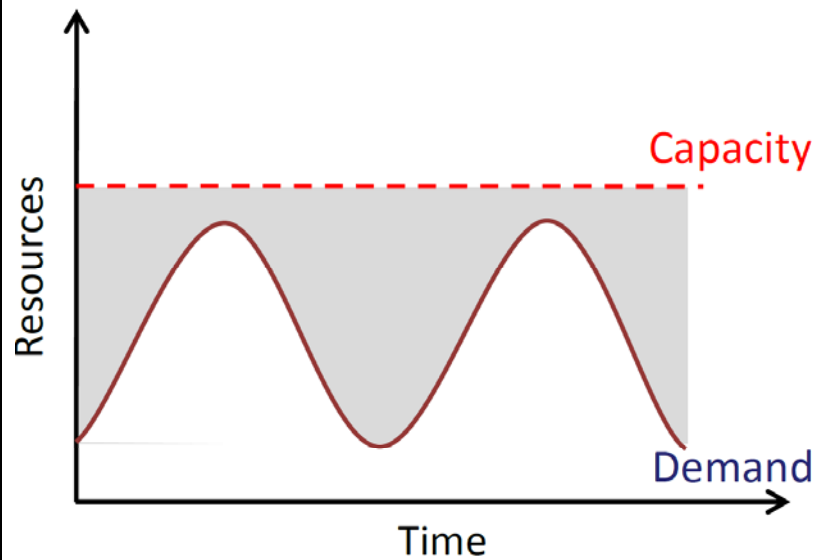


Computation as a utility

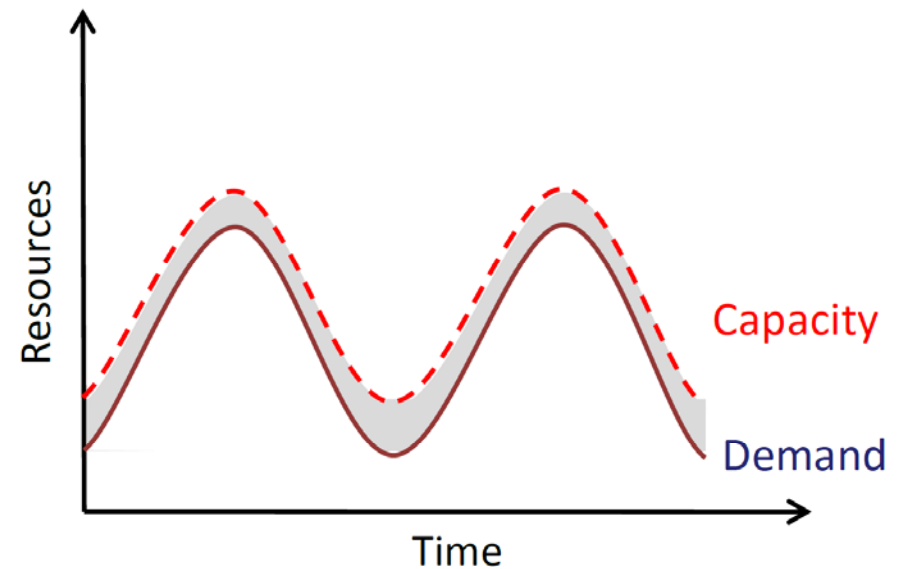


Management improvement

Private data center



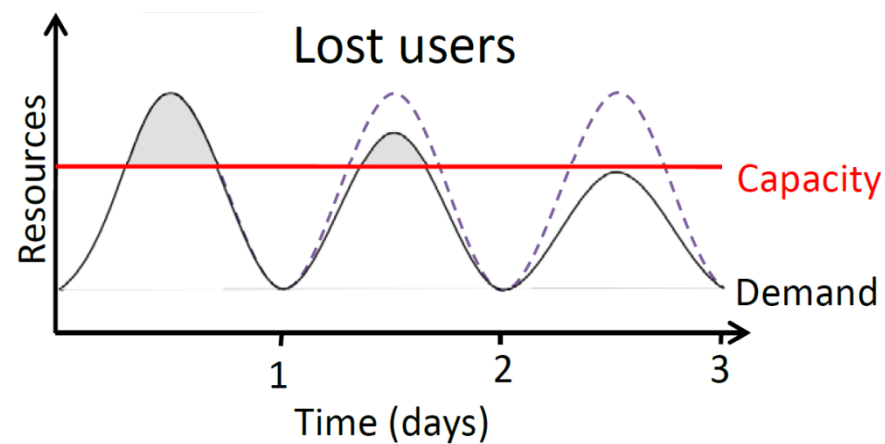
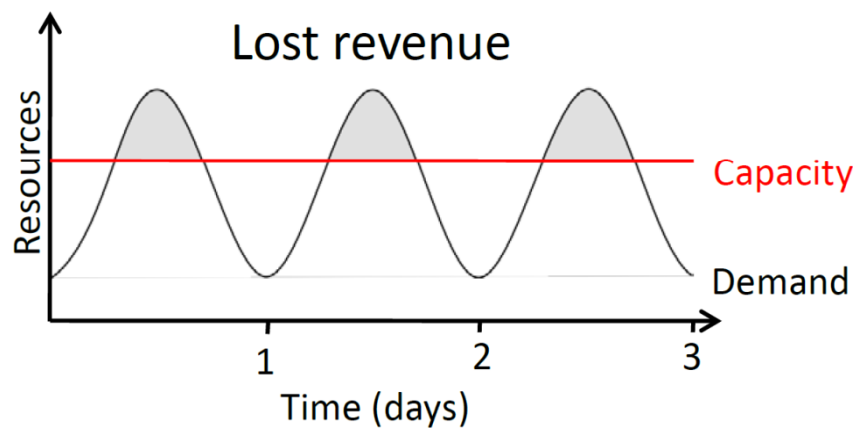
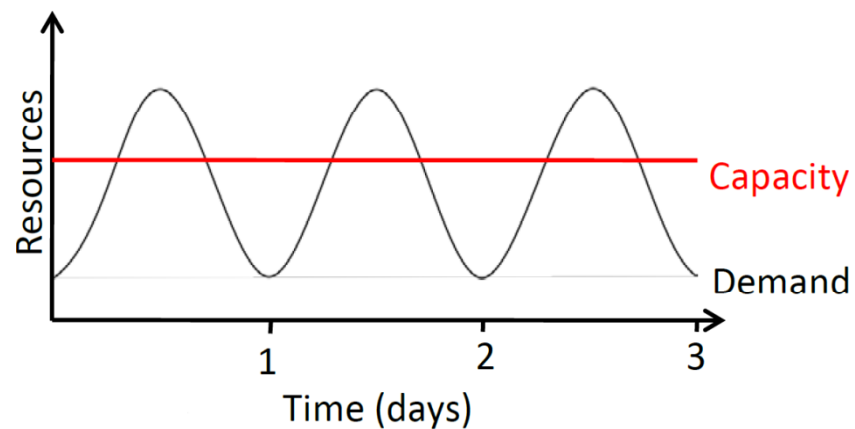
Public cloud



Daniel Abady, UC Berkeley



Undercapacity risk



Daniel Abady, UC Berkeley



Cloud computing definition

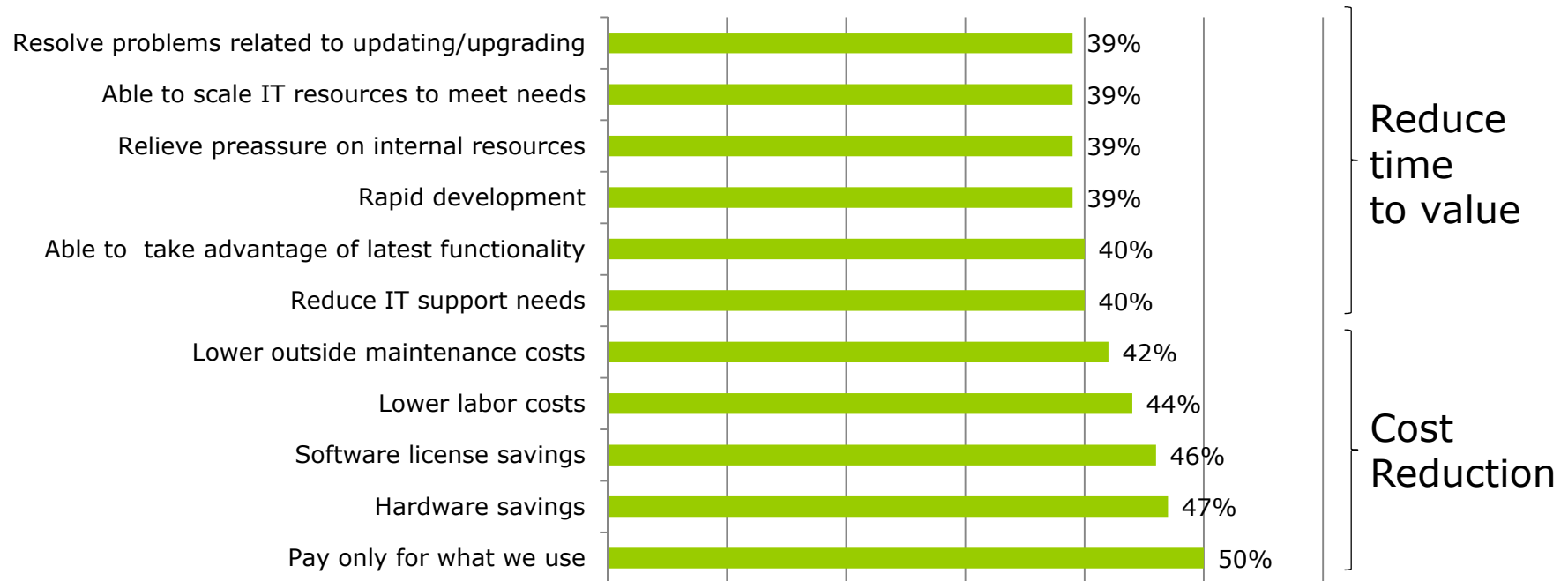
“Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

NIST (National Institute of Standards and Technology)



Benefits of Cloud Computing (I)

Benefits for deploying in a cloud environment



IBM global survey of IT and line-of-business decision makers 2012



Benefits of cloud computing (II)

- ❑ Reduce costs
- ❑ Energetic efficiency
- ❑ Flexibility
- ❑ Agility
- ❑ Superior safety
- ❑ Economy of scale in software development
- ❑ Better upgradeability
- ❑ Easier management
- ❑ More business
- ❑ Big Data



Characteristics

- ❑ On-demand self-service
- ❑ Broad network access
- ❑ Resource pooling
- ❑ Rapid elasticity
- ❑ Measured service



What's new

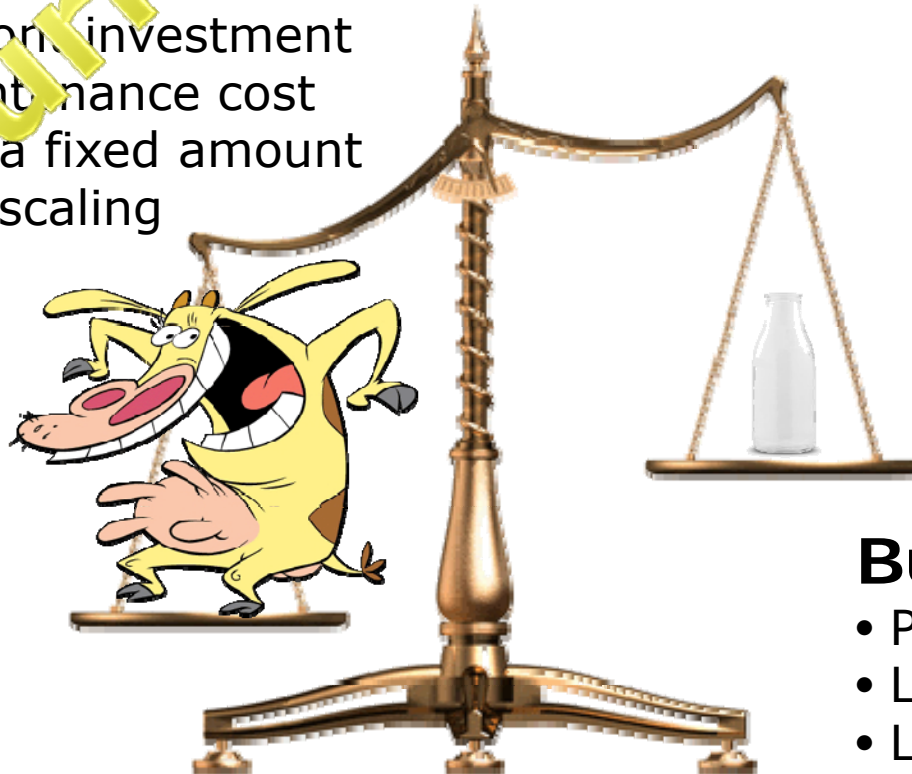
- ❑ Illusion of infinite resources
- ❑ Elimination of up-front commitment
- ❑ Pay-per-use
 - Cost is 5-7 times cheaper than in-house computing



A cow or bottled milk?

Buy a cow

- High upfront investment
- High maintenance cost
- Produces a fixed amount
- Stepwise scaling



Buy bottled milk

- Pay-per-use
- Lower maintenance cost
- Linear scaling
- Fault tolerant

Daniel Abadi analogy



Scalability/Elasticity

- ❑ It is a design/configuration issue
 - Current systems do not tell you *how many* CPUs or servers you must use
 - You cannot model what you cannot formally define

- ❑ There is not clear definition
 - The Universal Scalability Law (USL)
 - ❑ It is a mathematical definition of scalability



Universal Scalability Law (I)

- It can model both Sw or Hw scalability
- The USL is defined as follows:

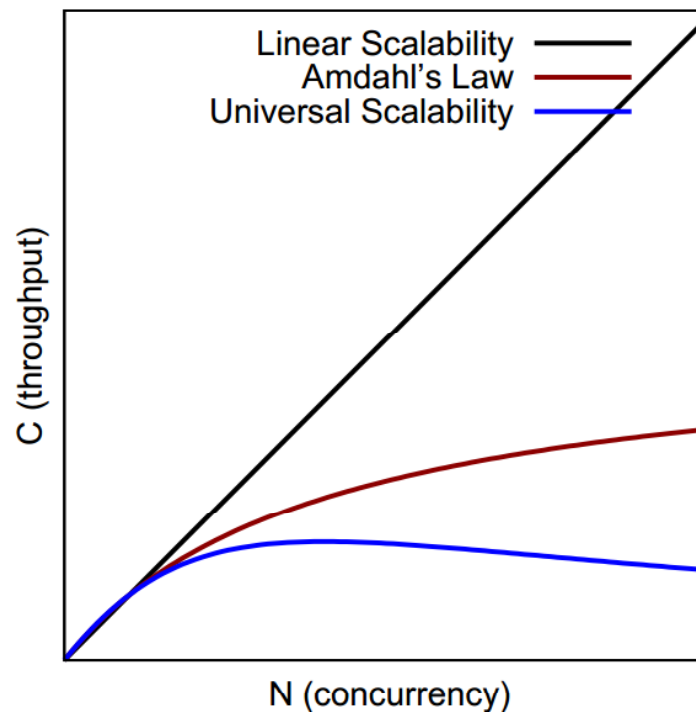
$$C(N) = \frac{N}{1 + \sigma(N - 1) + \kappa N(N - 1)}$$

- C: System's capacity (i.e., throughput) improvement
 - E.g., increment of number of queries per second
- N: System's size
 - (Sw): Number of concurrent threads
 - (Hw): Number of CPUs
- σ : System's contention
 - Performance degradation due to serial instead of parallel processing
- κ : System's consistency delay (aka coherency delay)
 - Extra work needed to keep synchronized shared data (i.e., inter-process communication)



Universal Scalability Law (II)

- If $\kappa = 0$, it simplifies to Amdahl's law
- If both $\sigma = 0$ and $\kappa = 0$, we obtain linear scalability



USL application method

1. Empirical analysis: Compute C (throughput) for different values of N (concurrency)
2. Perform least-squares regression against gathered data
 - a) $x := N - 1$
 - b) $y := N/C(N) - 1$ (being $C(N) = C_N/C_1$)
 - c) Fit a second-order polynomial of the form
$$y = ax^2 + bx + 0$$
3. Reverse the transformation
 - a) $\sigma = b - a$ (contention)
 - b) $\kappa = a$ (consistency delay)

<http://www.percona.com/files/white-papers/forecasting-mysql-scalability.pdf>



USL application method: Example

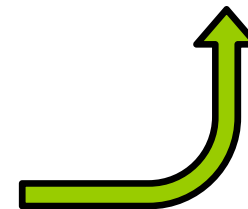
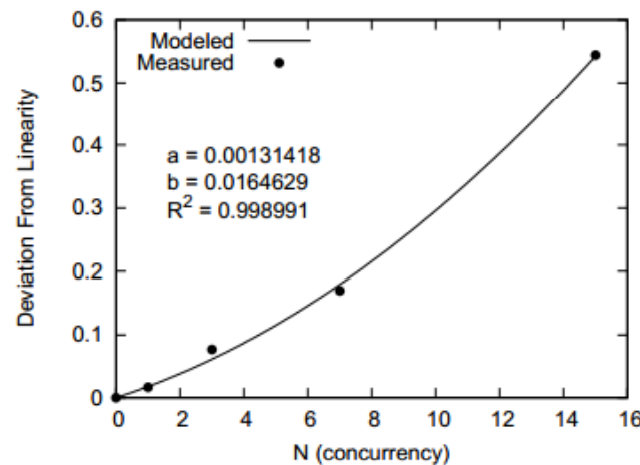
Concurrency (N)	Throughput (C)
1	955.16
2	1878.91
4	3548.68
8	6531.08
16	9897.24



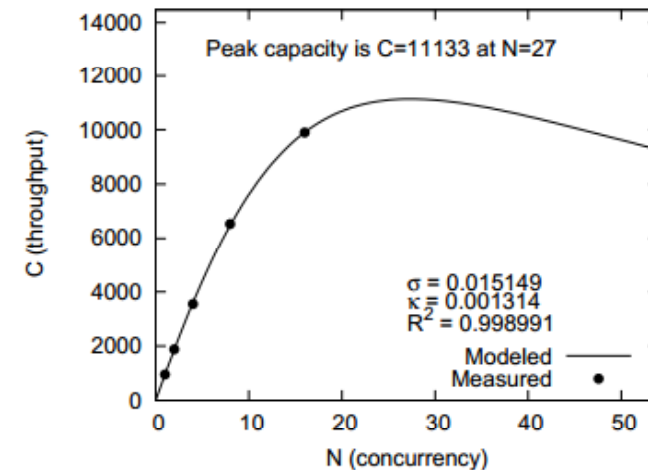
N	C	$N - 1$	$\frac{N}{C/C(1)} - 1$
1	955.16	0	0.0000
2	1878.91	1	0.0167
4	3548.68	3	0.0766
8	6531.08	7	0.1699
16	9897.24	15	0.5441



Points are fit in a second-order polynomial: ax^2+bx+0 , and a and b are computed



σ and κ are next computed from a and b ; and we can plot the USL function



Measuring Scalability

- ❑ Scalability is normally measured in terms of speed-up and scale-up
 - Speed-up: Measures performance when adding Hw for a constant problem size
 - ❑ Linear speed-up means that N sites solve in T/N time, a problem solved in T time by 1 site
 - Scale-up: Measures performance when the problem size is altered with resources
 - ❑ Linear scale-up means that N sites solve a problem $N \cdot T$ times bigger in the same time 1 site solves the same problem in T time
- ❑ The USL shows that linear scalability is hardly achievable
 - σ (contention) could be avoided (i.e., $\sigma = 0$) if our code has no serial chunks (everything parallelizable)
 - κ (consistency delay) could be avoided (i.e., $\kappa = 0$) if replicas can be synchronized without sending messages



Providers' Challenges

□ Deployment

- Localization
- Routing
- Authentication

□ Tuning

- Placement (mapping virtual to physical machines)
- Resource partitioning (scheduling)
- Service level objectives
- Dynamically varying workloads



Obstacles/Opportunities

- ❑ Availability of service
- ❑ Data lock-in
- ❑ Data confidentiality
- ❑ Data transfer bottlenecks
- ❑ Performance unpredictability
- ❑ Scalable storage
- ❑ Debugging
- ❑ Scaling quickly
- ❑ Reputation fate sharing
- ❑ Software licensing

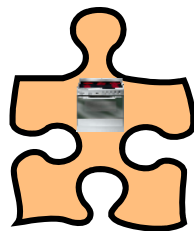


RaaS (visionary scenario)

- ❑ **Finer granularity**
 - Resources (e.g., CPU, RAM, I/O, etc.)
 - ❑ Also special resources like GPUs
 - Time units (i.e., per second)
- ❑ **Priority queues of consumers**
 - Depending on the price
- ❑ **Research challenges**
 - Economical
 - ❑ Subletting
 - Technological
 - ❑ Customer placement and migration

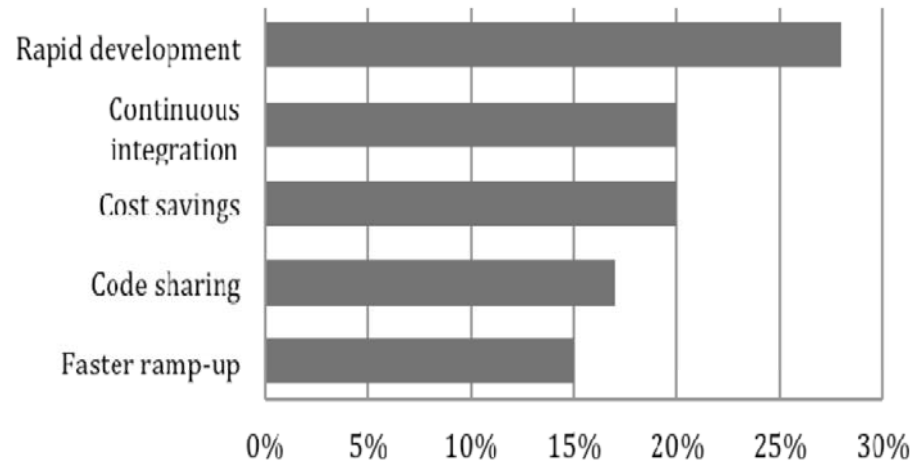


PaaS

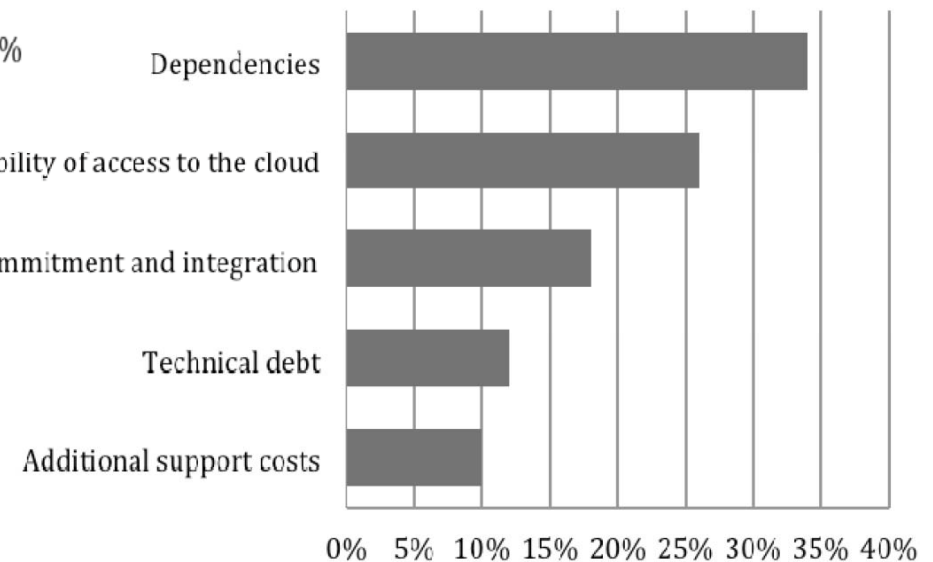


Distributed Software Development

Key benefits of using cloud in DSD



Key risks for DSD in the cloud

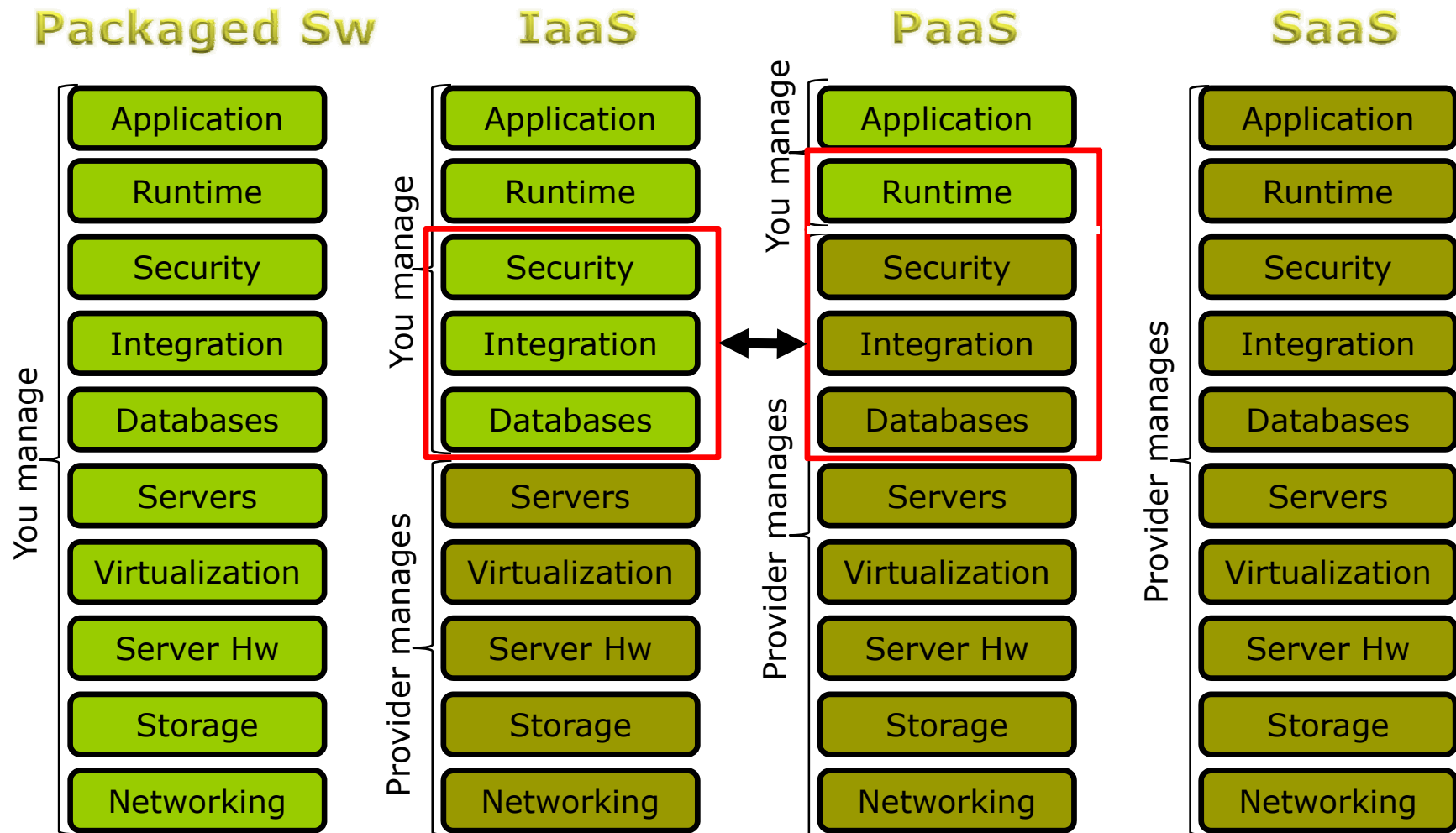


Kinds of platforms

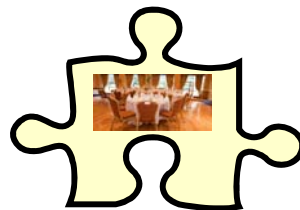
- ❑ Storage as a Service
- ❑ Database as a Service
- ❑ Processes (Workflow) as a Service
- ❑ Service as a Service
- ❑ Security as a Service
- ❑ Management/Governance as a Service



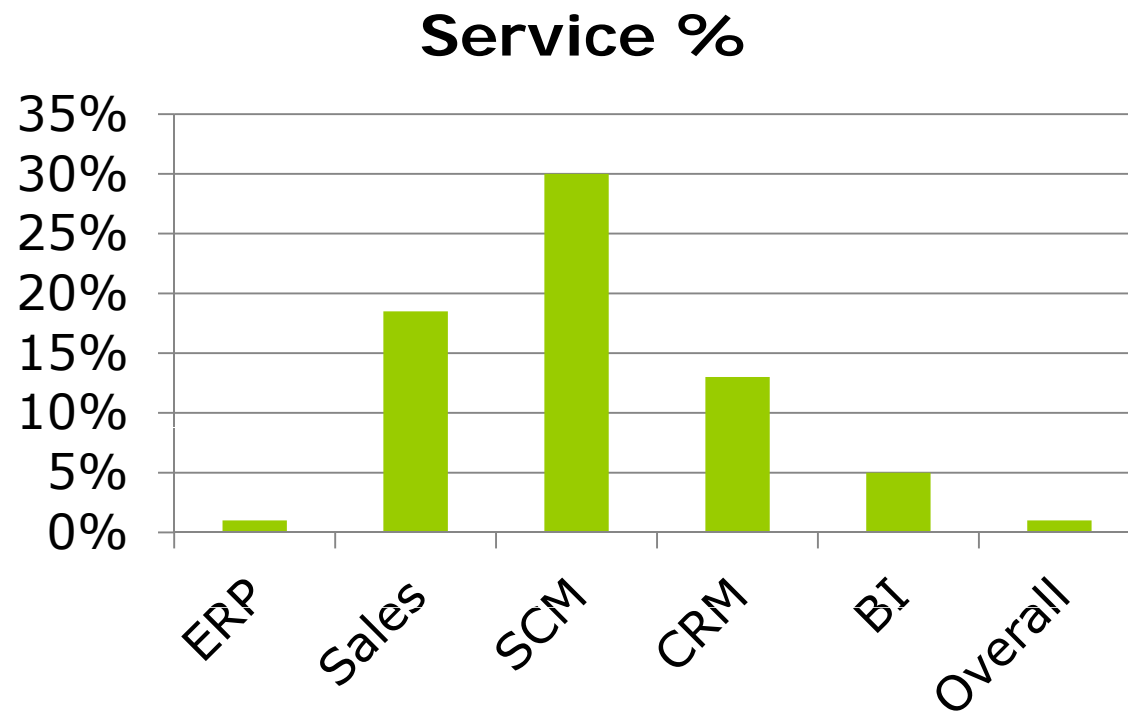
Responsibility



SaaS



Gartner's market analysis



Gartner's considerations on SaaS

	On-premises	Service-based
Customization	+	+/-
Implementation time	+/-	+/-
Application shut-off	+	-
Hidden fees	-	-
Security of data	+	-
Process integrity	+	-
Guarantee of quality	+	-



Multi-tenancy

“... a single instance of the software runs on a server, serving multiple client-organizations (tenants).”

Wikipedia

❑ Benefits:

- Improves efficiency
- Simplifies management

❑ Implementations:

- Shared hardware
 - ❑ Each tenant is assigned his/her own VM where the DBMS resides
 - Easy implementation
 - Strong isolation
 - High overhead
 - Redundant components and lack of coordination
- Shared database instance (i.e., process)
 - ❑ Effective resource sharing
 - ❑ Natively supported isolation at DBMS level
 - ❑ Workloads from independent tenants contend for the shared resources
 - No elasticity support
- Shared tables
 - ❑ All tenants share the same DBMS version and functionalities



Sharing modes

Sharing mode	Isolation	IaaS	PaaS	SaaS
Hardware	Virtual Machine	✓		
Virtual Machine	OS user	✓	✓	
Operating System	DB instance		✓	
Instance	Database		✓	
Database	Schema		✓	✓
Table	Row		✓	✓



Tenants management difficulties

- ❑ Variable popularity
- ❑ Unpredictable load characteristics
- ❑ Flash crowds
- ❑ Variable resource requirements



Multi-tenancy challenges

- ❑ Supporting thousands of tenants
 - Maintain metadata about customers (e.g., activated features)
 - Scale-out is necessary (sooner or later)
 - ❑ Rolling upgrades one server at a time
- ❑ Tolerating failures
- ❑ Self-managing
- ❑ Elastic load balancing
 - Dynamic partitioning of databases



Activity

- ❑ *Objective: Consider a dashboard as a service*
- ❑ *Tasks:*
 1. (5') *Individually make four lists considering a dashboard as a service*
 - a) *Benefits for the provider*
 - b) *Problems for the provider*
 - c) *Benefits for the consumer*
 - d) *Problems for the consumer*
 2. (10') *Explain your lists to the others*
 3. (10') *Merge the lists into only four*
 4. *Hand in two consensuated lists*
- ❑ *Roles for the team-mates during task 2:*
 - a) *Explains his/her material*
 - b) *Asks for clarification of blur concepts*
 - c) *Mediates and **controls time***



Summary

- ❑ Cloud computing definition
- ❑ Economical benefits of IaaS
- ❑ What's new in cloud computing
- ❑ Opportunities for using IaaS in BI
- ❑ PaaS benefits
- ❑ Universal Scalability Law
- ❑ Market situation of SaaS
- ❑ Multi-tenancy



Bibliography

- ❑ P. Mell and T. Grance. *The NIST Definition of Cloud Computing*. Special Publication 800-145, National Institute of Standards and Technology (January 2011), draft
- ❑ NIST Cloud Computing Program, <http://www.nist.gov/itl/cloud>
- ❑ C. Baun et al. *Cloud Computing*. Springer, 2011
- ❑ D. Abadi. *Data management in the cloud: Limitations and opportunities*. IEEE Data Engineering Bulletin 32(1), 2009
- ❑ C. Baun et al. *Cloud Computing*. Springer, 2011
- ❑ M. Madsen. *Cloud Computing Models for Data Warehousing*. Third Nature Technology White Paper, 2012
- ❑ N. J. Gunther. *A Simple Capacity Model of Massively Parallel Transaction Systems*. CMG National Conference, 1993
- ❑ O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, D. Tsafir. *The rise of RaaS: the resource-as-a-service cloud*. Commun. ACM 57 (7), 2014
- ❑ B. Hostmann. *Business Intelligence as a Service: Findings and Recommendations*. Research G00164653. Gartner, 2009
- ❑ K. Laudon and J. Laudon. *Management information systems : managing the digital firm*. Pearson/Prentice Hall, 2010
- ❑ K. Pauwels. *Dashboards as a Service: Why, What, How, and What Research Is Needed*. In JSR, August 2009



Resources

- ❑ aws.amazon.com/redshift
- ❑ aws.amazon.com/es/simplifiedb
- ❑ aws.amazon.com/es/free
- ❑ developers.google.com/prediction
- ❑ developers.google.com/storage
- ❑ developers.sap.com
- ❑ www.coherentpaas.eu
- ❑ www.google.com/drive/apps.html
- ❑ dashboardspy.wordpress.com
- ❑ www.quadrigram.com

